

Polzuweisung und Störungsentkopplung in linearen Systemen – ein numerisch orientierter Ansatz

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Bremen

vorgelegt von
Arno Linnemann
aus Bremerhaven

Bremen
1984





Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41543123_0036

POLZUWEISUNG UND STÖRUNGSENTKOPPLUNG
IN LINEAREN SYSTEMEN -
EIN NUMERISCH ORIENTIERTER ANSATZ

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Bremen

vorgelegt von
ARNO LINNEMANN
aus Bremerhaven

Bremen

1984

Die Dissertation lag dem Promotionsausschuß Dr.-Ing./
Dr.rer.nat. des Fachbereichs 3 der Universität Bremen vor.
An dem Promotionsverfahren haben Prof. Dr. D. Hinrichsen
(Bremen) und Prof. Dr. M.L.J. Hautus (Eindhoven) als Gut-
achter mitgewirkt. Das Kolloquium fand am 29.6.1984 statt.

Diese Arbeit ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Forschungsschwerpunkt Dynamische Systeme der Universität Bremen entstanden.

Mein besonderer Dank gilt Herrn Professor D. Hinrichsen für die Betreuung der Arbeit. Seine zahlreichen Hinweise und kritischen Anmerkungen waren mir eine große Hilfe.

Ferner konnte ich von zahlreichen Diskussionen mit Gastprofessoren des Forschungsschwerpunktes profitieren. In diesem Zusammenhang möchte ich mich besonders bei Professor M.L.J. Hautus, Professor A.J. Laub und Professor J.C. Willems bedanken.

Schließlich danke ich Herrn M. Meijers für die Unterstützung bei den Rechnerarbeiten, Frau E. Sieber für die sorgsam und zügig ausgeführten Schreibarbeiten und meiner Frau für ihre wertvolle Hilfe bei der Endredaktion der Arbeit.

INHALT

1. EINLEITUNG

- 1.1. Numerische lineare Systemtheorie 1
- 1.2. Die Zweischrittmethode für Systeme ohne Struktur 2
- 1.3. Zusammenfassung und Einordnung in die vorhandene Literatur 5

2. PRÄLIMINARIEN 14

- 2.1. Bezeichnungen 14
- 2.2. Kondition von Problemen, Stabilität von Algorithmen und orthogonale Transformationen 18

3. ZUSTANDSDARSTELLUNGSFORMEN 24

- 3.1. Systeme mit einem Eingang 27
 - 3.1.1. Hessenberg-Systeme 28
 - 3.1.2. Newton-Systeme 43
 - 3.1.3. Jacobi-Systeme 54
- 3.2. Systeme mit mehreren Eingängen 64
 - 3.2.1. Hessenberg-Systeme 64
 - 3.2.2. Newton-Systeme 79
 - 3.2.3. Jacobi-Systeme 81
- 3.3. Anmerkungen 91

4. POLZUWEISUNG DURCH ZUSTANDSRÜCKFÜHRUNG 93

- 4.1. Zustandsrückführung bei Systemen mit einem Eingang 94
 - 4.1.1. Ein Algorithmus für Frobenius-Systeme 95
 - 4.1.2. Ein Algorithmus für Newton-Systeme 96
 - 4.1.3. Eine Weiterentwicklung 98
 - 4.1.4. En bloc Algorithmen 102
 - 4.1.5. Numerische Erfahrungen 110
- 4.2. Zustandsrückführung bei Systemen mit mehreren Eingängen 113
 - 4.2.1. Sukzessive Algorithmen 114
 - 4.2.2. En bloc Algorithmen 119
- 4.3. Großsysteme mit dünn besetzten Systemmatrizen 124
- 4.4. Anmerkungen 128

5. STÖRUNGSENTKOPPLUNG 130

- 5.1. Geometrische Theorie der Störungsentkopplung 132
- 5.2. Analyse von Systemen mit Hilfe von Matrix-Algorithmen 141
 - 5.2.1. Verhältnis zwischen den Unterraum-Algorithmen 142
 - 5.2.2. Ein Matrix-Algorithmus 144
 - 5.2.3. Analyse von Systemen mit Algorithmus MAIU 157
- 5.3. Synthese von Systemen mit Hilfe von Matrix-Algorithmen 167
- 5.4. Matrizielle und strukturelle Charakterisierungen der Lösbarkeit 202

6. WAS IST FEEDBACK - WAS IST FEEDFORWARD ? 222

7. LITERATUR 257

ANHANG: Schlüssel zum Auffinden von Textstellen 264



1. EINLEITUNG

1.1. Numerische lineare Systemtheorie

Zur Analyse und Synthese von linearen endlichdimensionalen zeitinvarianten Regelungssystemen existieren sehr gut gearbeitete Theorien und Ansätze (beispielsweise (*Rosenbrock* 1970; *Wonham* 1974; *Wolovich* 1974)). Ferner ist es durch die sich rapide entwickelnde Computertechnologie möglich, die in der Systemtheorie entwickelten Konzepte auf immer größere Systeme anzuwenden. Dabei stellt sich jedoch ein Mangel der herkömmlichen Systemtheorie heraus. Viele der vorhandenen - theoretisch korrekten - Algorithmen versagen, wenn sie von einem Computer in endlicher Arithmetik ausgeführt werden. Dieses liegt daran, daß sich die vom Computer begangenen Rundungsfehler im Laufe einer umfangreichen Rechnung extrem "aufschaukeln" können, so daß das Endresultat erstaunlich stark von dem korrekten Ergebnis abweicht.

Dieses Phänomen ist bei Problemen der linearen Algebra schon seit langem bekannt (*Wilkinson* 1965) und hat zur Entwicklung eines neuen Gebietes, der *numerischen linearen Algebra* geführt. Das Ziel dieses Gebiets ist es, Algorithmen für Probleme der linearen Algebra zu entwickeln, die auch in endlicher Arithmetik zuverlässige Resultate liefern (*Wilkinson* und *Percekh* 1971). Erste Arbeiten, die der "numerischen linearen Systemtheorie" zuzuordnen wären, sind vor zirka 10 Jahren publiziert worden, und die Anzahl der in diesem Gebiet pro Jahr veröffentlichten Arbeiten nimmt ständig zu (vgl. etwa der Übersichtsartikel

(Laub 1983) und die Literatursammlung am Ende dieser Arbeit). Im Gegensatz zu Problemen der linearen Algebra (Smith u.a. 1976; Dongarra u.a. 1979) gibt es jedoch noch keine Computerprogramme zu Fragestellungen der linearen Systemtheorie, deren numerische Zuverlässigkeit allgemein anerkannt ist: Das Gebiet der numerischen linearen Systemtheorie ist erst im Aufbau begriffen.

Die vorliegende Arbeit liefert einen weiteren Beitrag zur Entwicklung dieses Gebiets, indem die Systemtheorie in solch eine Richtung weiterentwickelt wird, daß schließlich darauf aufbauende, numerisch zuverlässige Algorithmen entworfen und implementiert werden können. Manch ein Resultat mag dabei auch für den "Theoretiker" interessant sein, dessen Hauptinteresse nicht den numerischen Fragestellungen gilt.

1.2. Die Zweischrithmethode für Systeme ohne Struktur

Der Leitfaden der vorliegenden Arbeit ist die "Zweischrithmethode" zur Analyse und Synthese von Regelungssystemen ohne ausnutzbare Struktur. Die Methode und ihre Anwendbarkeit wird im folgenden kurz beschrieben.

Nimmt man an, daß die das System definierenden Matrizen keine ausnutzbaren Symmetrieeigenschaften besitzen (Hamiltonsche Systeme, symmetrische Systeme etc.), so kann man die beiden folgenden Systemklassen unterscheiden:

- (a) die definierenden Matrizen sind voll besetzt, d.h.
sie haben keine oder nur wenig verschwindende Einträge
(*Systeme ohne Struktur*)
- (b) die definierenden Matrizen sind dünn besetzt, d.h.
viele Einträge sind Null (*Systeme mit Struktur*).

In der numerischen linearen Algebra wird diese Unterscheidung ebenfalls getroffen und die numerischen Techniken für (a) und (b) sind sehr verschieden. Die folgende (unvollständige) Aufzählung der Methoden und Hilfsmittel zum Lösen von linearen Gleichungssystemen verdeutlicht dieses:

für voll besetzte Matrizen:

LR-Zerlegung, QR-Zerlegung, Singuläre-Werte-
Zerlegung, trianguläre Gleichungssysteme;

für dünn besetzte Matrizen:

Jacobi-Methode, Gauß-Seidel-Methode, Graphentheorie.

Bei dünn besetzten Matrizen stehen iterative Verfahren und kombinatorische Methoden im Vordergrund, während bei voll besetzten Matrizen immer wieder die folgende *Zweischritt-methode* auftaucht: In einem ersten Schritt wird das Problem in ein äquivalentes, aber einfacheres Problem transformiert, und im zweiten Schritt wird dann das einfachere Problem gelöst. So ist es beispielsweise üblich, die ein lineares Gleichungssystem definierende Matrix zunächst zu triangulieren, um dann das trianguläre Gleichungssystem zu lösen. Ein weiteres

Beispiel ist die Eigenwertberechnung. Hier wird die Matrix zunächst in eine Hessenberg-Matrix transformiert und dann der QR-Algorithmus auf die Hessenberg-Matrix angewendet. So findet man in der Standard-Referenz (*Wilkinson* 1965) auch die folgenden Überschriften zum Problem der Eigenwertberechnung: "Reduction of a general matrix to condensed form" (1. Schritt) und "Eigenvalues of matrices of condensed forms" (2. Schritt).

Bei der Zweischrittmethode läßt sich i.a. nicht ausnutzen, daß das Ausgangssystem dünn besetzt ist. So besitzt beispielsweise das im zweiten Schritt zu lösende Problem häufig keine zusätzliche Struktur mehr. Für dünn besetzte Systeme sind spezielle Verfahren, die die Struktur der Matrizen ausnutzen, oftmals sehr viel effizienter, und daher können mit deren Hilfe meist auch hochdimensionale Probleme gelöst werden.

Auch in der Systemtheorie sollten sich die Methoden für Systeme mit und ohne Struktur unterscheiden. Die bisher in der numerischen linearen Systemtheorie vorgestellten Algorithmen beziehen sich fast ausschließlich auf voll besetzte Probleme. Bei den Resultaten über "large scale systems" steht der numerische Aspekt nur bedingt im Vordergrund. Auch in dieser Arbeit wird (bis auf eine Ausnahme) angenommen, daß die zu untersuchenden Systeme keine Struktur besitzen, also in die Kategorie (a) einzuordnen sind.

Es stellt sich heraus, daß man auch in der Systemtheorie die Zweischrittmethode bereits praktiziert. Wird beispielsweise zur Polzuweisung das System zunächst in eine *Kontrollierbarkeitsnormalform* transformiert und mit dieser Normalform die Rückführung berechnet, so ist dieses eine Zweischrittmethode. Die Systeme, die im ersten Schritt des Zweischrittverfahrens erzeugt werden, müssen nicht notwendig Normalformen sein, es genügen Zustandsdarstellungsformen, für die das betreffende Regelungsproblem einfach lösbar ist. Häufig wird man jedoch Normalformen antreffen, da diese besonders wenig Parameter enthalten. Die bis vor kurzem benutzten Kontrollierbarkeitsnormalformen haben allerdings den Nachteil, daß sie sich nicht numerisch zuverlässig berechnen lassen.

Damit die Zweischrittmethode sich aber auch zu einem numerisch attraktiven Instrument entwickelt, sind für gegebene Regelungsprobleme dazugehörige Zustandsdarstellungsformen anzugeben, die sich einerseits *numerisch zuverlässig* berechnen lassen, und für die andererseits das Regelungsproblem *vereinfacht* wird. Die vorliegende Arbeit liefert Resultate in diese Richtung für diverse Versionen des Polzuweisungs- und Störungsentkopplungsproblems.

1.3. Zusammenfassung und Einordnung in die vorhandene Literatur

Das anschließende Kapitel 2 dient der Bereitstellung von *Notationen* und einiger *Grundbegriffe* aus der numerischen linearen Algebra. Im Kapitel 3 werden spezielle *Zustands-*

darstellungsformen angegeben, in die sich jedes erreichbare (bzw. beobachtbare) System durch einen Basiswechsel im Zustandsraum transformieren läßt. Das Kapitel 4, in dem die numerischen Aspekte der *Polzuweisung durch Zustandsrückführung* behandelt werden, baut sehr stark auf Kapitel 3 auf. Im Kapitel 5 werden diverse Versionen des *Störungsentkopplungsproblems* studiert. Obwohl die angewendeten Methoden in den Kapiteln 3, 4 und 5 weitgehend identisch sind, ist das Kapitel 5 doch größtenteils unabhängig von den Kapiteln 3 und 4. Das Kapitel 6 befaßt sich mit der Frage, wie *Feedback und Feedforward* zu definieren sind. Es ist aus Kapitel 6 hervorgegangen und benutzt Konzepte aus Kapitel 5, ist aber von unabhängigem Interesse.

Im folgenden wird auf die Kapitel 3 bis 6 im einzelnen eingegangen. Der Inhalt wird zusammengefaßt und in die vorhandene Literatur eingeordnet.

zu Kapitel 3

Die besondere Bedeutung von Zustandsdarstellungsformen, bei denen das System eine besonders einfache Struktur annimmt, wurde in der systemtheoretischen Literatur schon früh erkannt - vgl. (Luenberger 1967) und die dort gegebenen Referenzen - und hat zur Entwicklung von diversen "kanonischen Formen" geführt. Zunächst wurde die *kontroll-kanonische Form* (oder "phase-variable canonical form") für Systeme mit einem Eingang eingeführt, bei der die Systemmatrix eine Frobeniusmatrix ist. Später wurde diese dann auf multivariable Systeme verallge-

meinert (Luenberger 1967; Popov 1972; u.a.). Während es bei skalaren Systemen jedoch im wesentlichen nur zwei Möglichkeiten gab, kanonische Formen zu erklären (die freien Parameter der Frobenius-Matrix stehen entweder in den Zeilen oder in den Spalten), gab es bei multivariablen Systemen eine Vielzahl von Möglichkeiten, vgl. Luenberger (1967) und Hinrichsen und Prätzel-Wolters (1983). Am besten erforscht sind dabei diejenigen kanonischen Formen, deren Strukturen durch die Kronecker- bzw. Hermite-Indizes (Hinrichsen und Prätzel-Wolters 1983) gegeben sind.

Da sich alle diese Formen aber nicht mit Hilfe eines numerisch stabilen Algorithmus (Stewart 1973) berechnen lassen, hat man in letzter Zeit alternative Formen entwickelt, die weniger Struktur besitzen (mehr freie Parameter), sich aber numerisch stabil berechnen lassen (Nour Eldin 1977; Varga 1981; Nour Eldin und Heister 1980; Konstantinov, Petkov und Christov 1980; van Dooren 1981; Patel 1981; Paige 1981). In (Konstantinov, Petkov und Christov 1981) und (Hinrichsen und Linnemann 1984) wird gezeigt, wie man mit Hilfe dieser Formen Normalformen unter der orthogonalen Ähnlichkeitsaktion erhalten kann. Im skalaren Fall gibt es im wesentlichen nur eine dieser Formen, bei der die Systemmatrix eine Hessenberg-Matrix ist. Im multivariablen Fall sind bislang die beiden Formen studiert, die auf Kronecker- bzw. Hermite-Indizes aufbauen.

In der vorliegenden Arbeit werden zunächst (Abschnitt 3.1.) Systeme mit einem Eingang behandelt, da sich die einzelnen

Zustandsdarstellungsformen für diesen Spezialfall sehr viel einfacher und übersichtlicher beschreiben lassen. Die Verallgemeinerung dieser Formen auf multivariable Systeme geschieht dann in Abschnitt 3.2. Ferner wird die historische Entwicklung umgekehrt, indem zunächst auf die numerisch zuverlässigen Hessenberg-Formen eingegangen wird (Abschnitte 3.1.1. und 3.2.1.). Als neues Resultat wird dabei in Abschnitt 3.2.1. eine sehr allgemeine Klasse von Hessenberg-Systemen, die sich numerisch stabil berechnen lassen, und die zu Normalformen unter der orthogonalen Ähnlichkeitsaktion führen, vorgestellt. Die bereits bekannten Kronecker- und Hermite-Hessenberg-Formen sind darin als Spezialfall enthalten.

Für Systeme mit einem Eingang wird in Abschnitt 3.1.2. aus der Hessenberg-Form die Newton-Form eines Systems entwickelt. Die Existenz einer Newton-Form ist bereits bekannt (*Eising* 1982); die Eindeutigkeitsaussage ist neu. Die Newton-Form enthält als Spezialfall die klassische "kontroll-kanonische Form". Mögliche Verallgemeinerungen der Newton-Form auf multivariable Systeme werden in Abschnitt 3.2.2. nur anhand von Beispielen angedeutet.

In den Abschnitten 3.1.3. (skalare Systeme) und 3.2.3. (multivariable Systeme) wird eine neue Klasse von Systemen, die der *Jacobi-Systeme*, eingeführt. Diese Systeme sind *strukturell symmetrisch* bezüglich der Eingänge und Ausgänge in dem Sinne, daß ihre Struktur, festgelegt etwa durch die Kopplungsstruktur des Signalflußdiagramms, symmetrisch ist. "Fast alle" erreichbaren und beobachtbaren Systeme lassen sich durch einen Basis-

wechsel im Zustandsraum in ein Jacobi-System überführen, und man erhält auch eine Normalform bezüglich dieser Aktion. Betrachtet man "nicht generische" Situationen, so geht die angenehme strukturelle Symmetrie verloren und man erhält die in (Bosgra und von der Weiden 1980) eingeführte Normalform. Es lohnt sich jedoch, die generische Situation getrennt zu betrachten, da die Beweise dabei sehr viel einfacher, elementarer und durchsichtiger werden. Ferner wird, für Systeme mit einem Eingang, ein direkter Zusammenhang zu Kettenbruchzerlegungen und minimalen partiellen Realisierungen (Kalman 1971) deutlich. Als Anwendung von Jacobi-Systemen wird ein Stabilisierungsproblem mit Hilfe von reduzierten Modellen behandelt und eine Normalform unter Ähnlichkeit und statischer Ausgangsrückführung (Hinrichsen und Prätzel-Wolters 1984) abgeleitet.

zu Kapitel 4

In diesem Kapitel über die Berechnung einer Zustandsrückführung zur Polzuweisung steht der numerische Aspekt sehr stark im Vordergrund, und es werden mehrere Polzuweisungsalgorithmen vorgestellt.

Fast alle bisher bekannten Algorithmen bauen auf speziellen Zustandsdarstellungsformen auf, wobei bis vor kurzem die "kontroll-kanonischen Formen" im Vordergrund standen (vgl. Literaturanmerkungen in (Ackermann 1977)). Da sich jedoch herausstellte, daß sich die "kontroll-kanonischen Formen" für höherdimensionale Systeme nicht zuverlässig berechnen lassen,

hat man versucht, sie durch *Hessenberg-Formen* zu ersetzen. Da die *Hessenberg-Formen* wesentlich weniger Struktur besitzen als die "kontroll-kanonischen Formen", hat sich dieses als eine nichttriviale Aufgabe herausgestellt und zu einer Reihe von Veröffentlichungen geführt (*Nour Eldin* 1974; *Eising* 1982; *Miminis* und *Paige* 1982; 1982a; *Petkov, Konstantinov* und *Christov* (1982)). Die o.g. Algorithmen werden als numerisch zuverlässig angesehen, es existieren jedoch noch keine entsprechenden Rundungsfehleranalysen. Lediglich in (*van Dooren* 1984) sind partielle Resultate für ein spezielles Polzuweisungsproblem vorhanden. In den Abschnitten 4.1. (Systeme mit einem Eingang) und 4.2. (Systeme mit mehreren Eingängen) der vorliegenden Arbeit werden eine Reihe von Polzuweisungsalgorithmen vorgestellt. Das Verhältnis der Algorithmen zueinander und zu bereits bekannten Algorithmen wird diskutiert. Es zeigt sich, daß es im wesentlichen zwei Typen von Algorithmen gibt: *sukzessive Algorithmen*, bei denen jeder Pol einzeln verschoben wird, und *en bloc Algorithmen*, bei denen die Gesamtheit der Pole in einem Schritt verschoben wird. Die sukzessiven Algorithmen haben den Vorteil, daß sie auch für multivariable Systeme zu einem rein algebraischen Verfahren führen. Bei den *en bloc* Algorithmen hingegen lassen sich durch den Einsatz von *Optimierungsmethoden* zusätzliche Anforderungen an den Entwurf (etwa Minimierung der Rückführverstärkungen und Eigenvektorenuzuweisung) leicht berücksichtigen. Insbesondere erhält man ein zu (*Cavin* und *Bhattacharyya* 1982; *Kautsky* und *Nichols* 1983) alternatives Verfahren zur *robusten Polzuweisung*.

Die bisher erwähnten Polzuweisungsalgorithmen gehen alle von Systemen ohne ausnutzbare Struktur aus. Für *Großsysteme mit dünn besetzten Systemmatrizen* (vgl. Abschnitt 1.2.) wird in Abschnitt 4.3. ein spezieller Algorithmus vorgeschlagen, der auf iterativen Verfahren zur Eigenwertberechnung bei dünn besetzten Matrizen beruht.

zu Kapitel 5

Für die meisten Versionen des Störungsentkopplungsproblems gibt es in der *geometrischen Theorie* bereits sehr zufriedenstellende theoretische Resultate (Wonham 1974; Willems 1981; 1982; Willems und Commault 1981). Insbesondere existieren Charakterisierungen der Lösbarkeit in Termen von invarianten Unterräumen, und es gibt ad hoc Algorithmen zur Berechnung der Regler. Numerisch zuverlässige Algorithmen zur Störungsentkopplung sind jedoch erst für die einfachste Version des Problems - Störungsentkopplung durch Zustandsrückführung - vorhanden (Aplevich 1979; van Dooren 1981). Nach einer kurzen Einführung in die Problemstellung und die Resultate der geometrischen Theorie in Abschnitt 5.1. wird in Abschnitt 5.2. gezeigt, daß sich der numerisch stabile Algorithmus von Aplevich (1979) und van Dooren (1981) universell zur *Analyse von Systemen* einsetzen läßt.

Der Algorithmus von Aplevich (1979) und van Dooren (1981) liefert eine spezielle Zustandsdarstellungsform, mit deren Hilfe die Berechnung einer störungsentkoppelnden Zustandsrückführung dann einfach ist (vgl. Zweischrittmethode aus Abschnitt 1.2.).

Für komplexere Probleme (etwa Fast-Störungsentkopplung) liefert diese Zustandsdarstellungsform aber nicht genügend Struktur, um *Entwurfsalgorithmen* direkt darauf aufbauen zu können. Daher werden in Abschnitt 5.3. Variationen und Erweiterungen dieses Algorithmus entwickelt, die dann zu Entwurfsalgorithmen führen. Allerdings sind einige dieser Algorithmen nicht mehr numerisch stabil.

In Abschnitt 5.4. wird gezeigt, daß die in den Abschnitten 5.2. und 5.3. konstruierten speziellen Zustandsdarstellungsformen in dem Sinne typisch für gewisse Störungsentkopplungsprobleme sind, daß sich mit deren Hilfe *Charakterisierungen der Lösbarkeit* ableiten lassen. Für den Fall des Störungsentkopplungsproblems durch Zustandsrückführung ergibt sich die Charakterisierung von *Bhattacharyya, Gomes und Howze (1983)*. Die *matriziellen Charakterisierungen* des Abschnittes 5.4. lassen sich vielleicht einmal - ähnlich zur geometrischen Theorie - zu einer matriziellen Theorie linearer Systeme ausbauen. Der Vorteil eines matriziellen Ansatzes ist, daß sich die Ergebnisse graphisch mit Hilfe von Signalflußdiagrammen interpretieren lassen (strukturelle Charakterisierungen), daß die Objekte oftmals direkt und numerisch stabil berechnet werden können und daß sich Entwurfsverfahren kanonisch anschließen.

zu Kapitel 6

Obwohl "*Feedback*" und "*Feedforward*" zum festen Vokabular der Systemtheorie gehören, und obwohl der Systemtheoretiker eine feste Vorstellung damit verbindet, existiert bislang noch keine

allgemeingültige Definition dieser Begriffe. Lediglich für statische Zustandsregler hat *Kalman* (1972) eine Definition vorgeschlagen. Im abschließenden Kapitel 6 dieser Arbeit wird eine zu der von *Kalman* (1972) alternative Definition für statische Regler vorgestellt. Es zeigt sich, daß mit dieser Definition ein störungsentkoppelnder Zustandsregler stets Feedforward ist, bzw. Feedforwardanteile besitzt.

2. PRÄLIMINARIEN

In Abschnitt 2.1. werden zunächst einige Bezeichnungen eingeführt, und in Abschnitt 2.2. werden einige Grundbegriffe aus der numerischen linearen Algebra rekapituliert .

2.1. Bezeichnungen

Für $n \in \mathbb{N} := \{1, 2, 3, \dots\}$ bezeichnet $\underline{n}$ die Menge $\underline{n} := \{1, 2, \dots, n\}$.

Matrizen

Falls nicht anders erwähnt, sind alle Matrizen reell.

Mit I oder I_n wird die $(n \times n)$ -Einheitsmatrix bezeichnet.

Für eine Matrix A (reell oder komplex) bezeichnet A^T die transponierte und A^H die konjugiert transponierte Matrix.

Mit $\text{im } A$ ist das Bild, und mit $\text{ker } A$ ist der Kern von A gemeint. Für einen Unterraum U bezeichnet $U^\perp$ das orthogonale Komplement. Ferner werden die folgenden Matrizenklassen definiert:

$$\text{Gl}(n) := \{S \in \mathbb{R}^{n \times n} \mid \det S \neq 0\};$$

$$\text{Ut}(n) := \{T \in \text{Gl}(n) \mid t_{ij} = 0 \text{ für } i > j\};$$

$$\text{Ut}^*(n) := \{T \in \text{Ut}(n) \mid t_{ii} = 1 \text{ für } i \in \underline{n}\};$$

$$\text{Lt}(n) := \{T \in \text{Gl}(n) \mid T^T \in \text{Ut}(n)\};$$

$$\text{Lt}^*(n) := \{T \in \text{Gl}(n) \mid T^T \in \text{Ut}^*(n)\};$$

$$\text{O}(n) := \{Q \in \text{Gl}(n) \mid Q^T Q = Q Q^T = I\}.$$

Für komplexe Matrizen werden $Gl(n, \mathbb{C})$, $Ut(n, \mathbb{C})$, $Ut^*(n, \mathbb{C})$, $Lt(n, \mathbb{C})$, $Lt^*(n, \mathbb{C})$ entsprechend definiert. Ferner sei

$$U(n, \mathbb{C}) := \{Q \in \mathbb{C}^{n \times n} \mid Q^H Q = Q Q^H = I\}$$

die Menge aller unitären Matrizen. Die Elemente aus $O(n)$ werden orthogonal genannt.

Normen

Für einen Vektor $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ und eine Matrix

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

werden die folgenden Normen benutzt:

$$|x|_1 := \sum_{i=1}^n |x_i|$$

$$|x|_2 := (x^T x)^{1/2}$$

$$|x|_\infty := \max\{|x_i| \mid i \in \underline{n}\}$$

$$|A|_1 := \max\left\{\sum_{i=1}^m |a_{ij}| \mid j \in \underline{n}\right\}$$

$$|A|_2 := \max\{\sqrt{s} \mid s \text{ ist Eigenwert von } A^T A\}$$

$$|A|_\infty := \max\left\{\sum_{j=1}^n |a_{ij}| \mid i \in \underline{m}\right\}$$

$$A|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2}.$$

Ferner wird die L_p -Norm einer L_p -Funktion f mit $\|f\|_p$ bezeichnet ($1 \leq p \leq \infty$):

$$\|f\|_p := \left(\int_D |f(t)|^p dt \right)^{1/p} \quad \text{für } 1 \leq p < \infty$$

$$\|f\|_\infty := \sup_{t \in D} |f(t)| \quad \text{für } p = \infty.$$

Hierbei bezeichnet D den Definitionsbereich von f und — falls f vektor- oder matrixwertig ist — $|\cdot|$ eine der oben eingeführten Normen.

Komplexe Zahlen

Für komplexe Zahlen $s \in \mathbb{C}$ bezeichnet $\operatorname{Re}(s)$ den Realteil und $\operatorname{Im}(s)$ den Imaginärteil. Um die Eigenwerte einer Matrix mit ihren Multiplizitäten berücksichtigen zu können, werden die folgenden Begriffe eingeführt. Es sei die Äquivalenzrelation $\sim$ auf $\mathbb{C}^n$ definiert durch

$s \sim \tilde{s} \Leftrightarrow$ es gibt eine Permutation π der Komponenten von $\tilde{s}$, so daß $s = \pi \tilde{s}$.

Die Äquivalenzklassen $\{\tilde{s} \in \mathbb{C}^n \mid s \sim \tilde{s}\}$ seien mit $[s]$, und der Quotient $\{[s] \mid s \in \mathbb{C}^n\}$ sei mit $\mathbb{C}_\sim^n$ bezeichnet. Ferner sei die Abbildung $\operatorname{spec}: \mathbb{C}^{n \times n} \rightarrow \mathbb{C}_\sim^n$ definiert durch $\operatorname{spec}(A) = [(s_1, \dots, s_n)]$, wobei $s_1, \dots, s_n$ die Eigenwerte von A sind. Die Klasse $\operatorname{spec}(A) \in \mathbb{C}_\sim^n$ wird das *Spektrum*

von A genannt. Um Spektren von zusammengesetzten Matrizen untersuchen zu können, wird für $\tilde{s} \in \mathbb{C}^l$ und $s \in \mathbb{C}^n$ die "Vereinigung" eingeführt:

$$[\tilde{s}] \cup [s] := \begin{bmatrix} \tilde{s} \\ s \end{bmatrix} \in \mathbb{C}_{\sim}^{n+l}.$$

Außerdem wird für $l \leq n$ die Bezeichnung $[\tilde{s}] \subset [s]$ benutzt, falls es ein $\hat{s} \in \mathbb{C}^{n-l}$ mit $[\tilde{s}] \cup [\hat{s}] = [s]$ gibt. Für $s = (s_1, \dots, s_n) \in \mathbb{C}^n$ und $\mathbb{C}_g \subset \mathbb{C}$ wird das Zeichen " $\subset$ " auch in dem folgenden Sinne benutzt:

$$[s] \subset \mathbb{C}_g: \Leftrightarrow s_1, \dots, s_n \in \mathbb{C}_g.$$

Für reelle Matrizen hat das Spektrum spezielle Eigenschaften. Daher wird mit $\mathbb{C}_{\text{sym}}^n$ die Menge aller $s = (s_1, \dots, s_n) \in \mathbb{C}^n$ bezeichnet, für die ein $a \in \mathbb{N}$ existiert mit

$$\begin{aligned} s_1, s_2, \dots, s_{2a} &\in \mathbb{R} \\ s_{2a+1}, s_{2a+2}, \dots, s_n &\in \mathbb{R} \\ s_{2i} &= s_{2i-1}^H \quad \text{für } i \in \underline{a}. \end{aligned}$$

Die Zahl $a = a(s)$ gibt die Anzahl der konjugiert komplexen Paare an und ist durch s eindeutig bestimmt.

Es sei jetzt die Teilmenge $\mathbb{C}_{\text{sym}/\sim}^n$ von $\mathbb{C}_{\sim}^n$ definiert durch

$$\mathbb{C}_{\text{sym}/\sim}^n := \{[s] \mid s \in \mathbb{C}_{\text{sym}}^n\}.$$

Dann gilt für $A \in \mathbb{R}^{n \times n}$ stets

$$\text{spec}(A) \in \mathbb{C}_{\text{sym}/\sim}^n.$$

2.2. Kondition von Problemen, Stabilität von Algorithmen und orthogonale Transformationen

Die Begriffe der Kondition und numerischen Stabilität spielen eine zentrale Rolle für die vorliegende Arbeit, und daher soll hier darauf eingegangen werden. Die Darstellung ist jedoch sehr kurz gehalten, da man ausführliche Abhandlungen dieses Themas in jedem Lehrbuch zur numerischen linearen Algebra (etwa (Stewart 1973; Stoer und Bulirsch 1976)) und in letzter Zeit sogar in der systemtheoretischen Literatur (Paige 1981; von Dooren 1981; Laub 1983) findet.

Viele Probleme der Numerik lassen sich so formulieren, daß eine Abbildung f an einer Stelle x des Definitionsbereichs auszuwerten ist. Geschieht dieses mit Hilfe eines Computers in endlicher Arithmetik, so wird das Ergebnis durch Rundungsfehler verfälscht. Eine weitere Fehlerquelle sind ungenaue Eingangsdaten. Diese können entweder durch Meßwerte von beschränkter Genauigkeit hervorgerufen werden, oder dadurch entstehen, daß exakte Daten gerundet werden, um sie im Computer darzustellen. Ist das Problem jetzt so beschaffen, daß das zu bestimmende Ergebnis $f(x)$ sehr sensitiv auf Störungen in den Eingabedaten reagiert, so wird es *schlecht konditioniert* genannt. Da i.a. die Eingabedaten nicht exakt gegeben sind, kann man nicht erwarten, ein schlecht konditioniertes Problem zuverlässig lösen zu können.

Die Implementierung eines Algorithmus zur Lösung des Problems (Auswertung von f an der Stelle x) auf einem Computer führt zu einer neuen Abbildung f^* , die an der Stelle x ausgewertet wird. Bei der Beurteilung der Zuverlässigkeit eines Algorithmus ist zu prüfen, inwieweit $f^*(x)$ von $f(x)$ abweicht. Dieses hat sich aber für viele Probleme als sehr schwierig herausgestellt, und daher ist man dazu übergegangen, über die Güte eines Algorithmus anhand seiner numerischen Stabilität zu entscheiden. Ein Algorithmus (gegeben durch f^*) heißt *numerisch stabil*, falls sich (für alle x) das berechnete Ergebnis $f^*(x)$ interpretieren läßt als exaktes Ergebnis $f(x^*)$ bezüglich leicht gestörter Eingabedaten x^* . Auch ein numerisch stabiler Algorithmus kann zu beliebig schlechten Ergebnissen führen. Er liefert jedoch gute Resultate, falls das Problem gut konditioniert ist.

Falls der Algorithmus daraus besteht, daß eine Ausgangsmatrix mit einer Folge von speziellen orthogonalen Matrizen, den sogenannten *Householder-Matrizen* (Stoer und Bulirsch 1976), multipliziert wird, so hat Wilkinson (1965) gezeigt, daß sich der Algorithmus numerisch stabil implementieren läßt. Genauer gesagt, gilt das folgende Resultat: Es sei $A \in \mathbb{R}^{n \times n}$ und $Q \in O(n)$, wobei Q das Produkt von k Householder-Matrizen ist. Das Problem bestehe daraus, $\tilde{A} := QA$ zu bestimmen. Dann (Wilkinson 1965, Abschn. 3.45) lassen sich die Rechnungen so organisieren, daß für das berechnete Produkt $\bar{A}$ eine Matrix $\bar{Q} \in O(n)$ und eine Matrix $E \in \mathbb{R}^{n \times n}$ so existiert, daß

$$\bar{A} = \bar{Q}(A+E) \quad (1)$$

$$|E|_F \leq k(1+12.36 \cdot \text{eps})^{k-1} \cdot 12.36 \cdot \text{eps} \cdot |A|_F,$$

wobei eps die Rechengenauigkeit bezeichnet. Damit ist die Matrix $\bar{A}$ exakt orthogonal links-äquivalent zu einer leicht gestörten Matrix $A+E$. Man beachte, daß nichts über das Verhältnis von $\bar{A}$ zu A ausgesagt ist. In der Tat können A und $\bar{A}$ beliebig voneinander abweichen. Falls Q so gewählt ist, daß gewisse Einträge in $\tilde{A}$ verschwinden, so können diese Einträge auch in $\bar{A}$ zu Null gesetzt werden, ohne daß die Abschätzung (1) sich ändert. Ferner gilt eine entsprechende Abschätzung auch für den Fall, daß A nicht quadratisch ist, daß A von links und von rechts mit einer orthogonalen Matrix multipliziert wird ($Q_1 A Q_2$), und daß A durch eine Ähnlichkeitstransformation ($Q^T A Q$) verändert wird. Für die Ähnlichkeitstransformation ist (1) beispielsweise zu ersetzen durch

$$\bar{A} = \bar{Q}^T (A+E) \bar{Q} \quad (2)$$

$$|E|_F \leq 2k(1+12.36 \cdot \text{eps})^{2k-2} 12.36 \cdot \text{eps} |A|_F,$$

vgl. (Wilkinson 1965, Abschn. 3.45).

Das obige Resultat zeigt, daß sich orthogonale Transformationen numerisch stabil ausführen lassen, falls sie sich als Produkt von Householder-Matrizen darstellen lassen. Weitere Vorteile orthogonaler Transformationen bestehen darin, daß durch sie die Kondition vieler Probleme nicht verschlechtert wird

(vgl. (Stewart 1973, Thm. 4.1) für lineare Gleichungssysteme und (Stewart 1973, Thm. 4.10) für Eigenwerte), und daß sie effektiv dazu eingesetzt werden können, gegebene Matrizen zu vereinfachen. Daher sind orthogonale Transformationen für die in der Einleitung beschriebene "Zweischrittmethod" besonders geeignet.

Zum Abschluß dieses Abschnittes wird eine Möglichkeit, wie sich Matrizen durch orthogonale Transformationen vereinfachen lassen, kurz beschrieben. Zu jeder Matrix $A \in \mathbb{R}^{n \times m}$, $n, m \geq 1$ gibt es $Q \in O(n)$, so daß

$$QA = \begin{bmatrix} A_1 \\ 0 \end{bmatrix}, \quad (3)$$

wobei A_1 vollen Zeilenrang besitzt. Ist also $A_1 \in \mathbb{R}^{r \times m}$, so gilt $\text{rg } A = \text{rg } A_1 = r \leq m, n$. Algorithmisch lassen sich Q und A_1 mit Hilfe der QR-Zerlegung oder Singulären-Werte-Zerlegung bestimmen (Stoer und Bulirsch 1976, Abschn. 4.8 und 6.7; Stewart 1973, Abschn. 5.3 und 6.6). Die Singuläre-Werte-Zerlegung liefert dabei die zuverlässigsten Resultate, da die singulären Werte gut konditioniert sind und sich zuverlässig berechnen lassen. Allerdings ist eine Singuläre-Werte-Zerlegung sehr aufwendig. Die QR-Zerlegung läßt sich schneller und auch numerisch stabil erstellen. Allerdings ist die QR-Zerlegung nicht notwendig gut konditioniert, und man erhält auch keine Informationen darüber, wie weit A_1 von einer Matrix mit kleinerem Rang entfernt ist. Es hat sich

jedoch gezeigt, daß die QR-Zerlegung in den meisten Fällen ebenso zuverlässige Resultate liefert wie die Singuläre-Werte-Zerlegung. Man beachte, daß je nach verwendeter Zerlegung A_1 noch zusätzliche Eigenschaften besitzt (Stufenform oder orthogonale Zeilen).

In den Algorithmen, die im Laufe der vorliegenden Arbeit vorgestellt werden, kommen Transformationen der Form (3) immer wieder vor. Falls in einem Algorithmus eine Matrix Q wie oben konstruiert werden soll, so wird die folgende Schreibweise benutzt:

- Schiebe die Zeilen von A nach oben mit einer Matrix $Q \in O(n)$:

$$QA = \begin{bmatrix} A_1 \\ 0 \end{bmatrix},$$

wobei $A_1 \in \mathbb{R}^{r \times m}$ vollen Zeilenrang besitzt.

Hierbei ist es nicht von Bedeutung, wie Q konstruiert wird (QR- oder Singuläre-Werte-Zerlegung). Ferner kann die Zahl r entweder schon eingeführt sein (etwa als Rang von A), oder erst durch diesen Schritt definiert werden. Völlig analog wird die folgende Schreibweise benutzt:

- Schiebe die Zeilen von A nach unten mit $Q \in O(n)$:

$$QA = \begin{bmatrix} 0 \\ \underbrace{A_2}_m \end{bmatrix} \}_{r},$$

wobei A_2 vollen Zeilenrang besitzt.

Die Zahlen am Rand einer Matrix geben stets die Dimensionen der Teilmatrizen an. Falls diese Dimensionen durch den Schritt des Algorithmus erst definiert werden (d.h. die Variable r noch nicht eingeführt war), so wird zusätzlich ein Doppelpunkt verwendet:

$$QA = \begin{bmatrix} O \\ \underbrace{A_2}_m \end{bmatrix} : r .$$

Orthogonale Transformationen von rechts werden entsprechend behandelt:

- Schiebe die Spalten von A nach links (bzw. rechts) mit $Q \in O(m)$:

$$AQ = [A_1 \ O] \quad (\text{bzw.} \quad AQ = [O \ A_2]),$$

wobei A_1 (bzw. A_2) vollen Spaltenrang besitzt.

3. ZUSTANDSDARSTELLUNGSFORMEN

3.1. Systeme mit einem Eingang	27
3.1.1. Hessenberg-Systeme	28
3.1.2. Newton-Systeme	43
3.1.3. Jacobi-Systeme	54
3.2. Systeme mit mehreren Eingängen	64
3.2.1. Hessenberg-Systeme	64
3.2.2. Newton-Systeme	79
3.2.3. Jacobi-Systeme	81
3.3. Anmerkungen	91

In diesem Kapitel werden Systeme der Form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{1}$$

oder

$$\begin{aligned}x^{k+1} &= Ax^k + Bu^k \\ y^k &= Cx^k\end{aligned}\tag{2}$$

untersucht, die durch ein Tripel

$(A,B,C) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n}$ beschrieben werden.

Dabei wird grundsätzlich angenommen, daß (A,B) erreichbar und (A,C) beobachtbar ist, d.h. es gilt

$$\text{rg} R(A,B) = \text{rg } O(A,C) = n,\tag{3}$$

wobei

$$R(A,B) := [B, AB, \dots, A^{n-1}B]\tag{4}$$

die Erreichbarkeitsmatrix, und

$$O(A,B) := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\tag{5}$$

die Beobachtbarkeitsmatrix des Systems bezeichnet.

Die Menge aller erreichbaren und beobachtbaren Systeme fester Dimensionen wird mit $\Sigma(n,m,p)$ bezeichnet:

$$\begin{aligned}\Sigma(n,m,p) := \{ (A,B,C) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n} \mid & \text{rg } R(A,B) \\ & = \text{rg } O(A,C) = n \}.\end{aligned}\tag{6}$$

Für viele Betrachtungen wird die Ausgangsmatrix C keine

Rolle spielen und kann daher vernachlässigt werden.

Die Menge $\Sigma(n,m)$ enthält dann alle erreichbaren Systeme:

$$\Sigma(n,m) := \{(A,B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \mid \text{rg } R(A,B) = n\}. \quad (7)$$

Stellt man die Systeme (1) und (2) bezüglich einer neuen Zustandsvariablen

$$\tilde{x}(t) = S^{-1}x(t) \quad \text{bzw.} \quad \tilde{x}^k = S^{-1}x^k, \quad S \in \text{Gl}(n), \quad (8)$$

dar, so geht das Tripel (A,B,C) in das Tripel $(S^{-1}AS, S^{-1}B, CS)$ über. Dieser Übergang bedeutet lediglich einen Basiswechsel und viele Systemeigenschaften (beispielsweise Erreichbarkeit, Beobachtbarkeit, Entkoppelbarkeit, Eingangs-/Ausgangsverhalten) bleiben erhalten. Falls es ein $S \in \text{Gl}(n)$ gibt mit

$$(A_2, B_2, C_2) = (S^{-1}A_1S, S^{-1}B_1, C_1S), \quad (9)$$

so werden (A_1, B_1, C_1) und (A_2, B_2, C_2) *ähnlich* genannt.

Falls man S orthogonal wählen kann, so werden die Systeme *orthogonal ähnlich* genannt. Die *Ähnlichkeitsaktion*

$$\begin{aligned} \text{Gl}(n) \times \Sigma(n,m,p) &\rightarrow \Sigma(n,m,p) \\ (S, (A,B,C)) &\mapsto (S^{-1}AS, S^{-1}B, CS) \end{aligned} \quad (10)$$

ist eine der am meisten studierten Gruppenaktionen in der linearen Regelungstheorie. Für numerische Zwecke kann man diese Aktion nutzen, um ein gegebenes System (A,B,C) in ein System $(S^{-1}AS, S^{-1}B, CS)$ zu transformieren, für das bestimmte Regelungsprobleme besonders einfach zu lösen sind (vgl.

die in der Einleitung beschriebene Zweischrittmethode). In welche Form das System zu transformieren ist, hängt jedoch sehr stark vom Regelungsproblem ab: Eine Form, bei der ein bestimmtes Regelungsproblem (z.B. Polzuweisung) besonders einfach wird, kann sich im Zusammenhang mit anderen Problemen (z.B. Störungsentkopplung) als nutzlos herausstellen. In diesem Kapitel werden mehrere Klassen von Systemen vorgestellt, in die sich erreichbare Systeme transformieren lassen. Da die Erreichbarkeit an diesen Systemen direkt ablesbar ist, werden diese Systeme dann wichtig werden, wenn die Erreichbarkeit zur Lösung des Regelungsproblems entscheidend ausgenutzt werden muß. Für die vorgestellten Systeme werden die Anwendungen entweder direkt diskutiert, oder sie werden in Kapitel 4 zur numerischen Lösung des Polzuweisungsproblems herangezogen.

Um die Parallelen zur numerischen linearen Algebra, speziell zu dem Buch von *Wilkinson* (1965), hervorzuheben, könnte dieses Kapitel auch "Reduction of a general system to condensed form" (vgl. *Wilkinson* (1965), Kapitel 6) heißen.

3.1. Systeme mit einem Eingang

Für die Abschnitte 1.1. und 1.2. spielt die Ausgangsmatrix keine Rolle. Daher werden zunächst Systeme $(A,b) \in \Sigma(n,1)$ betrachtet.

3.1.1. Hessenberg-Systeme

Zunächst wird die Klasse der Hessenberg-Systeme definiert, dann werden einige Eigenschaften von Hessenberg-Systemen abgeleitet und schließlich wird gezeigt, daß sich jedes erreichbare System durch (9) in ein Hessenberg-System transformieren läßt.

Definition 1

Ein System $(A,b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n$ heißt *Hessenberg¹⁾-System*, falls

$$A = \begin{bmatrix} h_{11} & \cdot & \cdot & \cdot & h_{1n} \\ & \ddots & & & \vdots \\ h_{21} & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ 0 & & & h_{n,n-1} & h_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} h_{10} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

wobei

$$h_{i,i-1} \neq 0 \quad \text{für alle } i \in \underline{n}.$$

Die Menge aller Hessenberg-Systeme wird mit $H(n,1)$ bezeichnet.

Bemerkung 1

Bei einem Hessenberg-System ist die A-Matrix eine nicht-reduzierte Hessenberg-Matrix (Stewart 1973, S. 274).

¹⁾ K. Hessenberg

Bemerkung 2

Es gilt

$$(A,b) \in H(n,1) \Leftrightarrow R(A,b) \in Ut(n),$$

und daher sind Hessenberg-Systeme stets erreichbar.

Proposition 1

Zu jedem $(A,b) \in \Sigma(n,1)$ gibt es ein

$$Q \in O(n) \quad \text{mit} \quad (Q^T A Q, Q^T b) \in H(n,1).$$

Bis auf das Vorzeichen der Spalten ist die Matrix Q eindeutig bestimmt.

Beweis

Wegen Bemerkung 2 und der Identität

$$Q^T R(A,b) = R(Q^T A Q, Q^T b)$$

folgt die Existenzaussage direkt aus der Existenz einer QR-Zerlegung von $R(A,b)$. Die Eindeutigkeitsaussage ergibt sich aus der entsprechenden Eindeutigkeitsaussage der QR-Zerlegung (Wilkinson 1965, S. 241).

□

Hieraus ergibt sich unmittelbar:

Folgerung 1

Es bezeichne $H_+(n,1)$ die Menge aller Hessenberg-Systeme mit $h_{i,i-1} > 0$ für alle $i \in \underline{n}$ (Bezeichnung von Def. 1). Ferner sei $h: \Sigma(n,1) \rightarrow H_+(n,1)$ diejenige Abbildung, die

jedem erreichbaren System $(A, b) \in \Sigma(n, 1)$ das eindeutig bestimmte Hessenberg-System $(A_H, b_H) \in H_+(n, 1)$ mit

$$(A_H, b_H) = (Q^T A Q, Q^T b), \quad Q \in O(n),$$

zuordnet. Dann ist h eine Normalform bezüglich der Aktion

$$O(n) \times \Sigma(n, 1) \rightarrow \Sigma(n, 1) \\ (Q, (A, b)) \mapsto (Q^T A Q, Q^T b).$$

□

Das System $(A_H, b_H) \in H_+(n, 1)$ heißt *System-Hessenberg-Form* von $(A, b) \in \Sigma(n, 1)$ ¹⁾.

Zur Berechnung der System-Hessenberg-Formen mit Hilfe eines Computers mit endlicher Arithmetik ist es nicht ratsam, die Erreichbarkeitsmatrix heranzuziehen. Das folgende Beispiel von Laub (1983) zeigt, daß bereits bei der Aufstellung der Erreichbarkeitsmatrix wichtige Informationen verlorengehen können.

Beispiel 1

Angenommen, die Maschinengenauigkeit sei $\varepsilon > 0$ und es sei $\varepsilon < \mu < \sqrt{\varepsilon}$, wobei μ in der Maschine darstellbar ist.

Dann ist das System (A, b) mit

$$A = \begin{bmatrix} 1 & \mu \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ \mu \end{bmatrix}$$

¹⁾ Obwohl nicht eindeutig bestimmt, wird im folgenden das Hessenberg-System aus Proposition 1 hier manchmal auch System-Hessenberg-Form genannt.

sicher als erreichbar zu erkennen ($\text{rg } [sI-A, b] = n \forall s \in \mathbb{C}$).

Die Erreichbarkeitsmatrix ist jedoch gegeben durch

$$R(A, b) = \begin{bmatrix} 1 & 1+\mu^2 \\ \mu & \mu \end{bmatrix},$$

so daß mit endlicher Rechengenauigkeit die Matrix

$$\tilde{R}(A, b) = \begin{bmatrix} 1 & 1 \\ \mu & \mu \end{bmatrix}$$

berechnet wird. Diese Matrix ist jedoch singulär!

Dieses falsche Ergebnis kommt zustande, da unnötigerweise μ^2 in den Rechnungen auftaucht. Ein weiteres Beispiel, das diesen Punkt ebenfalls hervorhebt, findet sich in (Paige 1981).

Beispiel 1 zeigt, daß man direkt mit (A, b) arbeiten sollte, ohne vorher Produkte von Matrizen etc. zu bilden. Der folgende Algorithmus leistet dieses:¹⁾

¹⁾ Es gibt sehr viele verschiedene Möglichkeiten, einen Algorithmus zu beschreiben. In der vorliegenden Arbeit wurde kein Wert darauf gelegt, die Algorithmen möglichst so zu beschreiben, wie sie später evtl. implementiert werden. Vielmehr wurde aus Gründen der Übersichtlichkeit - und um die Notationen für einen evtl. anschließenden Induktionsbeweis bereitzustellen - eine Form gewählt, die weitestgehend einem mathematischen Beweis entspricht. So werden in dem folgenden Algorithmus beispielsweise orthogonale Matrizen explizit gebildet, was sich bei einer evtl. anschließenden Implementierung vermeiden läßt. Auch würden bei einer Implementierung die Matrizen $A_0, A_1, A_2, \dots$ etc. in einem Feld gespeichert werden, welches dann ständig überschrieben wird.

Algorithmus SHF

1. Eingabe $n \geq 1$ und $(A, b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1}$.
2. $k := 0$, $Q_0 := I \in O(n)$, $A_0 := A$, $b_0 := b$.
3. Falls $b = 0$, so stoppe $((A, b) \notin \Sigma(n, 1))$.
4. Schiebe die Zeilen von b nach oben mit $\bar{Q}_1^T \in O(n)$:

$$\bar{Q}_1^T b = [h_{10} \ 0 \dots 0]^T,$$

wobei $h_{10} \neq 0$.

5. $k \leftarrow k+1$.

6. Bilde

$$\tilde{Q}_k := \begin{bmatrix} I & 0 \\ 0 & \bar{Q}_k \end{bmatrix} \in O(n).$$

7. Bilde

$$A_k := \tilde{Q}_k^T A_{k-1} \tilde{Q}_k$$

$$b_k := \tilde{Q}_k^T b_{k-1}$$

$$Q_k := Q_{k-1} \tilde{Q}_k.$$

8. Kommentar: Es gilt

$$(i) \quad A_k = Q_k^T A Q_k, \quad b_k = Q_k^T b, \quad Q_k \in O(n).$$

- (ii)

$$A_k = \begin{bmatrix} * & \dots & * & & & \\ & h_{21} & & & & \\ & & \ddots & & & \\ & 0 & & h_{k,k-1} & & \\ & & & & a_{k+1,k} & \\ & & & & & \vdots \\ & 0 & & & & a_{n,k} \\ & & & & & & * & \dots & * \end{bmatrix}, \quad b_k = \begin{bmatrix} h_{10} \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

wobei $h_{i,i-1} \neq 0$ für $i \in \underline{k}$.

9. Falls $k = n$, so stoppe.

10. Falls $(a_{k+1,k}, \dots, a_{n,k})^T = 0$, so stoppe $((A,b) \notin \Sigma(n,1))$.

11. Schiebe die Zeilen von $(a_{k+1,k}, \dots, a_{n,k})^T$ nach oben mit

$\bar{Q}_{k+1}^T \in O(n-k)$:

$$\bar{Q}_{k+1}^T \begin{bmatrix} a_{k+1,k} \\ \vdots \\ a_{n,k} \end{bmatrix} = \begin{bmatrix} h_{k+1,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

wobei $h_{k+1,k} \neq 0$.

12. Gehe zu 5.

Bemerkung 3

Der Algorithmus SHF wurde von vielen Autoren (wieder-) entdeckt und zur Analyse und Synthese von Systemen vorgeschlagen (Munro und McLeod 1971; Aplevich 1974; Nour Eldin 1974; Tse, Medanic und Perkins 1978; Varga 1979; Konstantinov, Petkov und Christov 1980; Nour Eldin und Heister 1980; Paige 1981; Patel 1981; van Dooren 1981). Allerdings beziehen sich diese Arbeiten auf multivariable Systeme ($m \geq 1$), und daher werden die detaillierten Literaturanmerkungen zur System-Hessenberg-Form nach Abschnitt 3.2.1. verschoben.

Bemerkung 4

Der Algorithmus SHF ist sehr ähnlich zu Standardalgorithmen, die eine quadratische Matrix in eine Hessenberg-Matrix transformieren (z.B. Stewart 1973, Algorithm 1.1). Er zeichnet sich aber dadurch aus, daß b zusätzlich vereinfacht wird. Durch Übergang zu der zusammengesetzten Matrix

$$\tilde{A} = \begin{bmatrix} d & c \\ b & A \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)},$$

wobei $d \in \mathbb{R}$ und $c \in \mathbb{R}^{1 \times n}$ beliebig sind, können viele dieser Algorithmen jedoch auch zur Berechnung der System-Hessenberg-Form herangezogen werden. Dieses liegt daran, daß bei den meisten Implementierungen (etwa bei dem Unterprogramm ORTHES in EISPACK (Smith u.a. 1976) die Transformationsmatrix die Form $\tilde{Q} = \begin{bmatrix} 1 & 0 \\ 0 & Q \end{bmatrix}$ besitzt. Damit gilt dann aber

$$\tilde{Q}^T \tilde{A} \tilde{Q} = \begin{bmatrix} d & cQ \\ Q^T b & Q^T A Q \end{bmatrix}, \quad (11)$$

und $(Q^T A Q, Q^T b)$ ist ein Hessenberg-System, falls alle Nebendiagonalelemente von $\tilde{Q}^T \tilde{A} \tilde{Q}$ von Null verschieden sind. Man beachte, daß in (11) die Ausgangsmatrix für einen skalaren Ausgang mit transformiert wird.

Bemerkung 5

Aufgrund der Überlegungen in Abschnitt 2.2. läßt sich der Algorithmus SHF *numerisch stabil* implementieren, d.h. das berechnete Hessenberg-System ist die System-Hessenberg-Form für ein leicht gestörtes Ausgangssystem. Benutzt man beispielsweise das Unterprogramm ORTHES aus EISPACK, so erhält man einen numerisch stabilen Algorithmus.

Es soll jetzt die Kondition der System-Hessenberg-Form untersucht werden. Die folgenden Beispiele zeigen, daß System-Hessenberg-Formen sehr schlecht konditioniert sein können.

Beispiel 2

Es sei

$$A = \begin{bmatrix} -1 & & & & 0 \\ & a & & & \\ & & \ddots & & \\ & & & -1 & \\ 0 & & & & a & 0 \end{bmatrix}, \quad b = \begin{bmatrix} a \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

wobei $0 < a < 1$. Stört man b zu $b(\varepsilon) := [a, 0, \dots, 0, \varepsilon]$, so erhält man für $\varepsilon = -a^n$ ein nicht erreichbares System, denn

$$[a^{n-1}, a^{n-2}, \dots, a^2, a, 1][b(-a^n), A] =$$

$$[a^{n-1}, a^{n-2}, \dots, a^2, a, 1] \begin{bmatrix} a & -1 & & & 0 \\ 0 & a & & & \\ \vdots & & \ddots & & \\ 0 & & & -1 & \\ -a^n & 0 & & & a & 0 \end{bmatrix} = 0.$$

Die Hessenberg-Form zu $(A, b(-a^n))$ wird also mindestens ein verschwindendes Nebendiagonalelement enthalten, und daher ist die Hessenberg-Form von (A, b) sehr schlecht konditioniert. Nimmt man etwa $a = 1/2$ und $n = 20$, so verursacht eine Störung der Größenordnung $a^n \sim 10^{-6}$ in (A, b) eine Abweichung von $1/2$ in der System-Hessenberg-Form.

Beispiel 3

Das folgende System erhält man in Abwandlung der Ostrovsky-Matrix (Klema und Laub 1980), siehe auch (Miminis und Paige 1982):

$$A = \begin{bmatrix} 1 & \dots & 1 \\ -1 & \ddots & \vdots \\ 0 & \ddots & 1 & 1 \\ & & -1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Für größere n ist dieses Hessenberg-System sehr nahe bei einem unkontrollierbaren System, denn für $\varepsilon = 2^{-(n-1)}$ gilt

$$(2^{-n+1}, 2^{-n+2}, \dots, 2^{-1}, 2^0) \begin{bmatrix} -1 & 1 & \dots & \dots & 1 \\ & -1 & \ddots & & \vdots \\ & & \ddots & 1 & 1 \\ & 0 & & & \\ \varepsilon & \dots & \varepsilon & -1+\varepsilon & -1+\varepsilon \end{bmatrix} = 0.$$

Für $n = 20$ verursacht also eine Störung der Größenordnung $2^{-(n-1)} \sim 2 \cdot 10^{-6}$ eine Störung der Ordnung 1 in der System-Hessenberg-Form. Damit ist auch diese System-Hessenberg-Form schlecht konditioniert.

Diese schlechte Kondition der obigen Beispiele wird auch bei den folgenden Experimenten am Computer deutlich, bei denen die System-Hessenberg-Form mit Hilfe des Unterprogramms ORTHES aus EISPACK, wie oben beschrieben, berechnet wurde.¹⁾

Experiment 1

Eingegeben wurde das System aus Beispiel 2 mit $a = 0.1$ und variierendem n . Sodann wurde die "zufällige" Matrix $Q \in O(n)$ bestimmt, die das System $\tilde{A} := \text{diag}(1, 2, \dots, n)$, $\tilde{b} := (0.1, 0.1, \dots, 0.1)^T$ in die System-Hessenberg-Form transformiert. Schließlich wurde die System-Hessenberg-Form von $(Q^T A Q, Q^T b)$ berechnet. Bei exakter Rechnung würde (bis auf

¹⁾ Alle in dieser Arbeit beschriebenen Rechnungen wurden auf der Rechanlage Siemens 7.880 der Universität Bremen mit einfacher Genauigkeit (7 signifikante Dezimalstellen) ausgeführt.

Vorzeichen) wieder (A,b) erscheinen. Die folgende Tabelle gibt die Größenordnung der maximalen Abweichung in $Q^T Q - I$ und $(A,b) - (\bar{A}, \bar{b})$ wieder, wobei $(\bar{A}, \bar{b})$ das berechnete Hessesberg-System bezeichnet.

	n = 4	n = 6	n = 8	n = 15
$Q^T Q - I$	$0.5 \cdot 10^{-6}$	$0.3 \cdot 10^{-6}$	$0.2 \cdot 10^{-6}$	$0.8 \cdot 10^{-6}$
$(A,b) - (\bar{A}, \bar{b})$	$0.3 \cdot 10^{-4}$	$0.5 \cdot 10^{-3}$	0.15	1.0

Hierbei trat für $n = 4, 6, 8$ die maximale Abweichung jeweils im $(n, n-1)$ -ten Eintrag der A-Matrix auf.

Experiment 2

Dieses Experiment war genauso angelegt wie Experiment 1, nur daß das System aus Beispiel 3 betrachtet wurde. Die folgende Tabelle zeigt die Ergebnisse.

	n = 4	n = 10	n = 15	n = 20
$Q^T Q - I$	$0.5 \cdot 10^{-6}$	$0.4 \cdot 10^{-6}$	$0.8 \cdot 10^{-6}$	$0.1 \cdot 10^{-5}$
$(A,b) - (\bar{A}, \bar{b})$	$0.2 \cdot 10^{-4}$	$0.3 \cdot 10^{-3}$	$0.2 \cdot 10^{-1}$	0.93

Der maximale Fehler trat dabei jeweils im $(n, n-1)$ -ten Eintrag der A-Matrix auf.

Bemerkung 6

Wird Algorithmus SHF auf ein nicht erreichbares System (A,b) angewendet, so konstruiert er $Q \in O(n)$ mit

$$Q^T A Q = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad Q^T b = \begin{bmatrix} b_1 \\ 0 \end{bmatrix}, \quad (12)$$

wobei $(A_{11}, b_1) \in H(k, 1)$ und $0 \leq k < n$. Der erreichbare Teil des Systems wird also abgespalten. Daher läßt sich der Algorithmus SHF auch als Erreichbarkeitstest und zur Konstruktion einer minimalen Realisierung verwenden. Dieses Vorgehen ist in (Munro und McLeod 1971; Nour Eldin 1977; Varga 1979; Nour Eldin und Heister 1980; Patel 1981a; van Dooren 1981; Varga 1981) - gleich für multivariable Systeme - erläutert. Man beachte jedoch, daß die Funktion $f: \mathbb{R}^{n \times n} \times \mathbb{R}^n \rightarrow \{0, 1\}$ mit

$$f(A, b) = \begin{cases} 1 & \text{falls } (A, b) \in \Sigma(n, 1) \\ 0 & \text{sonst} \end{cases}$$

unstetig und daher extrem schlecht konditioniert ist. Falls jedoch $Q^T b$ und die ersten k Spalten von $Q^T A Q$ in (12) gut konditioniert sind, so ist ein nicht erreichbares System an kleinen Nebendiagonalelementen zu erkennen. Daher kann man die Unstetigkeit von f durch Einführung von "Nullschränken" (Hinrichsen und Linnemann 1984) kompensieren. Das Problem einer möglichen schlechten Kondition der System-Hessenberg-Form bleibt jedoch bestehen. Das folgende Experiment zeigt jedoch, daß System-Hessenberg-Formen "in der Regel" gut konditioniert sind.

Experiment 3

Untersucht wurden nicht erreichbare Systeme der Form

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n-2,1} & \dots & a_{n-2,n} \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{bmatrix},$$

wobei die Koeffizienten im Intervall $[-1,1]$ mit Hilfe des in (Smith u.a. 1976) vorgeschlagenen Zufallszahlengenerators erzeugt wurden. Ausgelesen wurde dann der Absolutbetrag des $(n,n-1)$ -ten Eintrages der A-Matrix in der System-Hessenberg-Form, der bei exakter Rechnung Null sein müßte. Die folgende Tabelle zeigt die Ergebnisse für 20 Experimente mit $n = 29$.

$ a_{n,n-1} $	$\infty - 10^{-3}$	$10^{-3} - 10^{-4}$	$10^{-4} - 10^{-5}$	$10^{-5} - 10^{-6}$	$10^{-6} - 0$
#					
Experimente	0	1	6	11	2

Bemerkung 7

Neben den bereits skizzierten Anwendungen (Erreichbarkeits-test und minimale Realisierung) findet die System-Hessenberg-Form noch Anwendungen bei der Polzuweisung durch Zustands- und Ausgangsrückführung (vgl. Kapitel 4), bei der Störungs-entkopplung und Berechnung von invarianten Unterräumen (vgl. Kapitel 5), bei einer Definition von Feedback und Feedforward

(vgl. Kapitel 6), beim Beobachterentwurf (*Nour Eldin* 1974a), bei der Modellreduktion (*Tse, Medanic* und *Perkins* 1978), bei der Aufstellung von Prim-Matrix-Produkten (*Nour Eldin* und *Heister* 1980; *Patel* 1981), bei der Auswertung von Transferfunktionen und der Berechnung von Frequenzantworten (*Varga* und *Sima* 1981; *Laub* und *Linnemann* 1984) und in der dezentralen Regelung (*Linnemann* 1984). Die Liste der Anwendungen nimmt ständig zu, und es ist zu erwarten, daß die System-Hessenberg-Form sich zum zentralen Objekt in der numerischen linearen Systemtheorie entwickeln wird.

Bemerkung 8

Falls das Ausgangssystem $(A,b,c) \in \Sigma(n,1,1)$ symmetrisch ist, d.h. falls $A = A^T$, $c^T = b$ gilt, so hat die System-Hessenberg-Form die Struktur

$$Q^T A Q = \begin{bmatrix} h_{11} & h_{21} & & & 0 \\ h_{21} & \ddots & \ddots & & \\ & \ddots & \ddots & h_{n,n-1} & \\ 0 & & h_{n,n-1} & h_{nn} & \\ & & & & \end{bmatrix}, \quad Q^T b = \begin{bmatrix} h_{10} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$cQ = [h_{10} \ 0 \ . \ . \ . \ . \ . \ . \ 0], \quad h_{1,i-1} \neq 0.$$

Dieses gilt, da orthogonale Ähnlichkeitstransformationen die Symmetrie nicht zerstören.

Im allgemeinen haben aber die nicht spezifizierten Einträge der System-Hessenberg-Form keine besonderen Eigenschaften.

Will man weitere Struktur erzwingen (etwa weitere Parameter eliminieren), so hat man nicht-orthogonale Transformationen auszuführen. Das folgende Lemma besagt, daß man in diesem Fall - bis auf die erste Zeile der A-Matrix - ein beliebiges Hessenberg-System erzeugen kann.

Lemma 1

Es sei $(A, b) \in \Sigma(n, 1)$ und $(\tilde{A}, \tilde{b}) \in H(n, 1)$, d.h.

$$\tilde{A} = \begin{bmatrix} h_{11} & \cdot & \cdot & \cdot & \cdot & h_{1n} \\ & h_{21} & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ 0 & & & & h_{n,n-1} & h_{nn} \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} h_{10} \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \quad h_{i,i-1} \neq 0. \quad (13)$$

Dann existiert eine eindeutig bestimmte Matrix $S \in GL(n)$ und eindeutig bestimmte Zahlen $a_1, a_2, \dots, a_n \in \mathbb{R}$ mit

$$S^{-1}AS = \begin{bmatrix} h_{11}-a_1 & \cdot & \cdot & \cdot & \cdot & h_{1n}-a_n \\ & h_{21} & h_{22} & \cdot & \cdot & h_{2n} \\ & & \cdot & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ 0 & & & & h_{n,n-1} & h_{nn} \end{bmatrix}, \quad S^{-1}b = \begin{bmatrix} h_{10} \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}. \quad (14)$$

Beweis

Die Existenz wird durch den im Anschluß an diesen Beweis vorgestellten Algorithmus sichergestellt. Zur Eindeutigkeit: Es seien (A_1, b_1) und (A_2, b_2) zu (A, b) ähnliche Systeme der Form (14). Dann gilt $b_1 = b_2$ und $A_1 + b_1 f = A_2$

für ein $f \in \mathbb{R}^{1 \times n}$. Da aber A_1 und A_2 das gleiche charakteristische Polynom besitzen, folgt $f = 0$, d.h. $A_1 = A_2$. Die Eindeutigkeit von S folgt aus $S^{-1}R(A,b) = R(\tilde{A},\tilde{b})$.

□

Der folgende Algorithmus führt ein beliebiges Hessenberg-System in die Form (14) über, wobei $(\tilde{A},\tilde{b})$ vorgegeben ist. In den folgenden Abschnitten werden für $(\tilde{A},\tilde{b})$ spezielle Systeme eingesetzt.

Algorithmus 1

1. Eingabe $n \geq 1$, $(A,b) \in H(n,1)$, $(\tilde{A},\tilde{b}) \in H(n,1)$
(bezeichne die Einträge von (A,b) mit a_{ij} und die Einträge von $(\tilde{A},\tilde{b})$ mit h_{ij}).

2. $k := n$, $A_n := A$, $b_n := b$, $S_n := I$.

3. Bilde $\bar{S}_n := \text{diag}(\frac{h_{10}}{a_{10}}, \frac{h_{21}h_{10}}{a_{21}a_{10}}, \dots, \frac{h_{n,n-1} \dots h_{10}}{a_{n,n-1} \dots a_{10}})$.

4. $k \leftarrow k-1$.

5. Bilde

$$\begin{aligned} A_k &:= \bar{S}_{k+1}^{-1} A_{k+1} \bar{S}_{k+1} \\ b_k &:= \bar{S}_{k+1}^{-1} b_{k+1} \\ S_k &:= S_{k+1} \bar{S}_{k+1}. \end{aligned}$$

6. Kommentar: Es gilt

$$(i) \quad A_k = S_k^{-1} A S_k, \quad b_k = S_k^{-1} b, \quad S_k \in \text{Gl}(n).$$

$$(ii) \quad A_k = \begin{bmatrix} * & \dots & \dots & \dots & \dots & * \\ h_{21} & \ddots & & & & \vdots \\ & \ddots & \ddots & & & \vdots \\ & & h_{k+1,k} & * & \dots & * \\ & & h_{k+2,k+1} & h_{k+2,k+2} & \dots & h_{k+2,n} \\ & 0 & & \ddots & \ddots & \vdots \\ & & & & h_{n,n-1} & h_{nn} \end{bmatrix}, \quad b_k = \begin{bmatrix} h_{10} \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

7. Falls $k = 0$, so stoppe.

8. Bilde

$$\bar{S}_k := \begin{bmatrix} 1 & \dots & \dots & 0 \\ & \ddots & & \\ & & 1 & \\ & & & 1 & s_{k+1} \dots s_n \\ & 0 & & 1 & \dots & 0 \\ & & & & \ddots & 1 \end{bmatrix},$$

wobei (für $i = k+1, k+2, \dots, n$) $s_i := \frac{h_{k+1,i} - *_{k+1,i}}{h_{k+1,k}}$, und

$*_{k+1,i}$ den $(k+1,i)$ -ten Eintrag in A_k bezeichnet.

9. Gehe zu 4.

Man beachte, daß sich Algorithmus 1 nicht numerisch stabil implementieren läßt.

3.1.2. Newton-Systeme

In diesem Abschnitt werden reelle Newton-Systeme (Koeffizienten in $\mathbb{R}$) und komplexe Newton-System (Koeffizienten in $\mathbb{C}$) behandelt. Vom praktischen Gesichtspunkt aus sind die

reellen Newton-Systeme sicherlich wichtiger, die Darstellung komplexer Newton-Systeme ist jedoch einfacher. Als Spezialfälle ergeben sich die Frobenius-Systeme (üblicherweise kontroll-kanonische Formen genannt) und die Jordan-Systeme. Die Systeme dieses Abschnitts werden in Kapitel 4 zur Polzuweisung herangezogen.

Definition 2

Es sei $s = (s_1, \dots, s_n)^T \in \mathbb{C}^n$. Ein (komplexes) System $(A, b) \in \mathbb{C}^{n \times n} \times \mathbb{C}^n$ heißt *komplexes Newton¹⁾-System mit Stützstellen* $s_1, \dots, s_n$, falls

$$A = \begin{bmatrix} s_1 - a_1 & -a_2 & \dots & -a_n \\ & 1 & & \\ & & s_2 & \\ & & & \ddots & 0 \\ & 0 & & 1 & \ddots & \\ & & & & \ddots & s_n \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \quad (15)$$

wobei $a_1, a_2, \dots, a_n \in \mathbb{C}$. Die Menge aller komplexen Newton-Systeme mit Stützstellen $s_1, \dots, s_n$ wird mit $N_S^{\mathbb{C}}(n, 1)$ bezeichnet.

Bemerkung 9

Die A-Matrix eines Newton-Systems ist eine "verallgemeinerte Begleitmatrix" im Sinne von *Werner* (1983). Für das charakteristische Polynom eines Newton-Systems kann man schnell nachrechnen, daß (*Werner* 1983)

$$\det(sI - A) = p_0(s) + a_1 p_1(s) + \dots + a_n p_n(s),$$

¹⁾ I. Newton (1643-1727)

wobei

$$p_i(s) := (s-s_{i+1}) \cdot \dots \cdot (s-s_n), \quad i = 0, 1, \dots, n-1,$$

$$p_n(s) := 1.$$

Daher sind die a_i in (15) die Koeffizienten der *Newtonschen Darstellung des charakteristischen Polynoms*.

In (Werner 1983) wurde gezeigt, daß sich jede komplexe Matrix durch (komplexe) Ähnlichkeitstransformationen in die in (15) gezeigte Form transformieren läßt. Da man sich leicht überzeugt, daß Lemma 1 auch über $\mathbb{C}$ gilt, erhält man die folgende Verschärfung dieses Resultats.

Proposition 2

Zu jedem erreichbaren System $(A, b) \in \mathbb{C}^{n \times n} \times \mathbb{C}^n$ und jedem $s = (s_1, \dots, s_n) \in \mathbb{C}^n$ gibt es genau eine invertierbare Matrix $S \in \mathbb{C}^{n \times n}$ mit

$$(A_N, b_N) := (S^{-1}AS, S^{-1}b) \in \mathbb{C}_S^n(n, 1).$$

Das System (A_N, b_N) wird (*komplexe*) *System-Newton-Form* von (A, b) genannt.

Bemerkung 10

Auch falls (A, b) reell ist und $s \in \mathbb{C}_{\text{sym}}^n$ (vgl. Abschn. 2.1.), so sind die Koeffizienten $a_1, \dots, a_n$ in (15) komplex. Bei dem folgenden reellen Newton-System wird komplexe Arithmetik gänzlich vermieden.

Definition 3

Es sei $s \in \mathbb{C}_{\text{sym}}^n$ und für $i \in \underline{a}$, $a_i = a(s)$, sei
 $(\tilde{s} - s_{2i})(\tilde{s} - s_{2i-1}) = \tilde{s}^2 + \alpha_i \tilde{s} + \beta_i$. Dann heißt $(A, b) \in \Sigma(n, 1)$
reelles Newton-System mit Stützstellen $s_1, \dots, s_n$, falls

$$A = \begin{bmatrix} -\alpha_1 - a_1 & -\beta_1 - a_2 & -a_3 & -a_4 & \dots & \dots & -a_n \\ 1 & 0 & & & & & \\ & 1 & -\alpha_2 & -\beta_2 & & & \\ & & 1 & 0 & & & \\ & & & 1 & \dots & 1 & -\alpha_a & -\beta_a \\ & & & & & 1 & 0 \\ & & & & & & 1 & s_{2a+1} \\ & & & & & & & 1 & \ddots \\ & & & & & & & & 1 & \ddots \\ & & & & & & & & & 1 & s_n \end{bmatrix} \quad (15a)$$

und $b = [1, 0, \dots, 0]^T$, wobei $a_1, a_2, \dots, a_n \in \mathbb{R}$. Die Menge
 aller reellen Newton Systeme mit Stützstellen $s_1, \dots, s_n$
 wird mit $N_s^{\mathbb{R}}(n, 1)$ bezeichnet.

Aus Lemma 1 folgt unmittelbar:

Proposition 3

Zu jedem $(A, b) \in \Sigma(n, 1)$ und jedem $s \in \mathbb{C}_{\text{sym}}^n$ gibt es genau
 ein $S \in \text{Gl}(n)$ mit $(S^{-1}AS, S^{-1}b) \in N_s^{\mathbb{R}}(n, 1)$.

Bemerkung 11

Eine Zustandsdarstellungsform eines multivariablen (!) Systems, die sehr ähnlich zu einem reellen Newton-System ist, wurde in (Eising 1982) zur Polzuweisung (Beweis und Algorithmus) herangezogen (vgl. auch Kapitel 4). Die Existenzaussage in Proposition 3 ist daher bereits bekannt; lediglich die Eindeutigkeitsaussage ist neu.

Als Spezialfall der Newton-Systeme erhält man die "kontrollkanonischen Formen", die hier Frobenius Systeme genannt werden.

Definition 4

Ein System $(A,b) \in \Sigma(n,1)$ heißt *Frobenius¹⁾-System*, falls $(A,b) \in N_O^R(n,1)$. Definiere $F(n,1) := N_O^R(n,1)$.

Bemerkung 12

Bei einem Frobenius System ist die A-Matrix eine Frobenius Matrix (Wilkinson 1965, Abschn. 1.13), die auch Begleit-Matrix genannt wird.

Aus Proposition 3 erhält man sofort:

Folgerung 2

Zu jedem $(A,b) \in \Sigma(n,1)$ gibt es genau ein $S \in Gl(n)$ mit $(A_F, b_F) := (S^{-1}AS, S^{-1}b) \in F(n,1)$.

Das System (A_F, b_F) heißt *System-Frobenius-Form* von (A,b) . In der Literatur wird dieses System auch "kontroll-kanonische Form" oder "phase-variable canonical form" genannt.

¹⁾G. Frobenius (1849-1917)

Bemerkung 13

Die numerischen Eigenschaften der Frobenius-Form einer einzelnen Matrix (Wilkinson 1965, Abschn. 6.52-6.56) gelten sinngemäß auch für die System-Frobenius-Form. Insbesondere läßt sich die System-Frobenius-Form nicht numerisch stabil berechnen, und die Kondition der Eigenwerte von A kann sich beim Übergang zu (A_F, b_F) drastisch verschlechtern.

Als weiteren Spezialfall erhält man die hier neu eingeführten Jordan-Systeme.

Definition 5

Ein (reelles oder komplexes) Newton-System heißt (reelles oder komplexes) $Jordan^1)$ -System, falls in (15) bzw. (15a) gilt: $a_1 = a_2 = \dots a_n = 0$. Die Menge aller Jordan-Systeme wird mit $Jor^{\mathbb{R}}(n,1)$ bzw. $Jor^{\mathbb{C}}(n,1)$ bezeichnet.

Bemerkung 14

Bei einem Jordan-System sind $s_1, s_2, \dots, s_n$ die Eigenwerte von A . Der Name "Jordan-System" wurde gewählt, da die A -Matrix Ähnlichkeit mit einer Jordan-Matrix (Wilkinson 1965, Abschn. 1.7) besitzt.

Aus den Propositionen 2 und 3 folgt sofort:

Folgerung 3

Zu jedem $(A,b) \in \Sigma(n,1)$ gibt es ein $S \in Gl(n)$ (bzw. $S \in \mathbb{C}^{n \times n}$ invertierbar) mit $(S^{-1}AS, S^{-1}b) \in Jor^{\mathbb{R}}(n,1)$

¹⁾C. Jordan (1838-1922)

(bzw. $\text{Jor}^{\mathbb{C}}(n,1)$). Bis auf Vertauschung der s_i sind diese *System-Jordan-Form* und S eindeutig bestimmt.

Mit Hilfe der folgenden Experimente sollte die Berechenbarkeit der System-Newton-, Frobenius- und Jordan-Formen mit einem Computer untersucht werden.

Experiment 4

Der Pseudozufallszahlengenerator aus (Smith u.a. 1976) wurde so modifiziert, daß er jeweils Zahlen aus dem Intervall $[-1,1]$ lieferte. Mit dessen Hilfe wurden dann jeweils eine zufällige Matrix $A \in [-1,1]^{n \times n}$ und zwei zufällige Vektoren $b_1, b_2 \in [-1,1]^n$ generiert. Sodann wurden mit Hilfe des Algorithmus SHF und des Algorithmus 1 die komplexen System-Newton-Formen von (A, b_1) und (A, b_2) bestimmt, bei denen bezüglich der Wahl der Stützstellen $s_1, s_2, \dots, s_n$ die folgenden Fälle unterschieden wurden.

1. Fall: $s_i = 0$ für alle $i \in \underline{n}$. Dieses ist die
(in reeller Arithmetik implementierte)
System-Frobenius-Form;
2. Fall: $s_i = -1$ für alle $i \in \underline{n}$;
3. Fall: $s_1, s_2, \dots, s_n$ für $n = 12$ wie in Bild 1 gewählt;



Bild 1

4. Fall: $s_1, s_2, \dots, s_n$ sind die durch den Algorithmus HQR aus EISPACK berechneten Eigenwerte von A (System-Jordan-Form).

5. Fall: $s_1, s_2, \dots, s_n$ sind die durch den Algorithmus HQR berechneten stabilen ($\text{Re } s_i \leq -0.5$) Eigenwerte von A zusammen mit komplexen Zahlen der Form $s_j = -0.5 \pm i\beta_j$, wobei die β_j die Imaginärteile der durch HQR berechneten instabilen Eigenwerte von A sind.

Bei exakter Rechnung müßten die Formen zu (A, b_1) und (A, b_2) jeweils übereinstimmen, und im 4. Fall müßten die Koeffizienten $a_1, a_2, \dots, a_n$ in (15) verschwinden.

Die folgende Tabelle gibt für ein Experiment mit $n = 12$ zunächst die Größenordnungen der Absolutbeträge der Koeffizienten $a_1, \dots, a_n$ für die Fälle 1, 2, 3 und 5 wieder.

Anzahl der Koeffizienten mit Betrag in

	$[0, 10^{-3}]$	$(10^{-3}, 10]$	$(10, 50]$	$(50, 10^2]$	$(10^2, 10^3]$	$(10^3, 10^4]$	$(10^4, 10^5]$	$(10^5, \infty)$
1. Fall		5	4	2	1			
2. Fall			1	1	4	6		
3. Fall	3	4	2	0	1	3	5	6
5. Fall	4	6	4	4	6			

Tabelle 1

Hierbei wurden im 3. und 5. Fall jeweils Real- und Imaginärteile gezählt. Man erkennt, daß die Koeffizienten sehr groß werden, wenn die Eigenwerte von A durch $s_1, \dots, s_n$ nicht relativ genau "getroffen" werden. Bild 2 gibt die Eigenwertverteilung der untersuchten Matrix A wieder.



Bild 2

Man beachte, daß die Größenordnungen der Einträge sich beim Übergang zur System-Hessenberg-Form nicht wesentlich ändern, da orthogonale Matrizen die Norm erhalten. Diese Tatsache wird auch durch Beobachtungen am Rechner bestätigt.

Die folgende Tabelle gibt die maximalen relativen und absoluten Abweichungen in den berechneten Koeffizienten $a_1, \dots, a_n$ der System-Newton-Formen von (A, b_1) und (A, b_2) für das obige Experiment mit $n = 12$ wieder.

	relative Abweichung	absolute Abweichung
1. Fall	$0.3 \cdot 10^{-4}$	$0.1 \cdot 10^{-2}$
2. Fall	$0.1 \cdot 10^{-5}$	$0.1 \cdot 10^{-1}$
3. Fall	0.5	1
5. Fall	1	$0.3 \cdot 10^{-2}$

Tabelle 2

Die großen relativen Abweichungen im 3. und 5. Fall kommen dabei durch ungenaue, kleine Koeffizienten zustande, und die absoluten Abweichungen sind den großen Koeffizienten zuzuschreiben. Man erkennt, daß es schon für $n = 12$ Schwierigkeiten bereitet, die System-Frobenius-Form zuverlässig zu berechnen.

Im 4. Fall lag der Betragsgrößte Koeffizient a_i bei $0.9 \cdot 10^{-2}$. Man beachte jedoch, daß in diesem Fall die Ungenauigkeiten teilweise auch durch fehlerhaft berechnete Eigenwerte und eine sensitive (bezüglich Änderungen in den s_i) System-Newton-Form hervorgerufen sein können.

Weitere Experimente mit $n = 12$ ergaben ähnliche Ergebnisse, und für zunehmendes n verstärkten sich die hier beschriebenen Effekte noch. So gab es bei $n = 20$ im 1. Fall beispielsweise sechs Koeffizienten größer als 10^4 , und im Fall 2 traten Größenordnungen von 10^8 auf. Entsprechend verschlechterte sich die absolute Abweichung. Die relative Abweichung blieb allerdings kleiner als 10^{-3} . Im 4. Fall trat bei $n = 20$ ein Koeffizient der Größenordnung 66 auf.

Zusammenfassend kann man sagen, daß bei der Berechnung der System-Newton-, Frobenius- und Jordan-Formen schon bei mittleren Dimensionen erhebliche Fehler auftreten. Ferner sind die Koeffizienten i.a. sehr groß. Inwiefern sich diese Fehler auf eventuell anschließende Entwurfsalgorithmen auswirken, wird in Kapitel 4 untersucht.

3.1.3. Jacobi-Systeme

In diesem Abschnitt werden durch $(A, b, c) \in \Sigma(n, 1, 1)$ gegebene Systeme betrachtet. Die Resultate dieses Abschnitts sind entstanden mit dem Ziel, eine Normalform für die Ähnlichkeitsaktion zu entwickeln, die möglichst symmetrisch bezüglich der Eingänge und Ausgänge des Systems ist.

Definition 6

Ein System $(A, b, c) \in \Sigma(n, 1, 1)$ heißt *Jacobi¹⁾-System*, falls

$$A = \begin{bmatrix} a_{11} & a_{12} & & 0 \\ 1 & & a_{22} & \\ & \ddots & \ddots & \\ & 0 & & 1 & \\ & & & & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad c = [a_{01} \ 0 \dots 0]$$

mit $a_{i-1,i} \neq 0$ für $i \in \underline{n}$. Die Menge aller Jacobi-Systeme wird mit $J(n, 1, 1)$ bezeichnet.

¹⁾ C.G.J. Jacobi (1804-1851)

Bemerkung 15

Die A-Matrix eines Jacobi-Systems ist tridiagonal, also eine Jacobi-Matrix (Gantmacher, Kapitel XIII, §8). Jacobi-Systeme sind *strukturell symmetrisch* bezüglich des Ein- und Ausgangs in dem Sinne, daß ihre Kopplungsstruktur, festgelegt etwa durch die Kopplungsstruktur des Signalflußdiagramms, symmetrisch ist.

Jacobi-Systeme haben ein sehr spezielles Eingangs- Ausgangs-Verhalten (z.B. ist $cb \neq 0$). Daher kann man nicht erwarten, daß jedes System zu einem Jacobi-System ähnlich ist. Es wird sich jedoch zeigen, daß "fast alle" Systeme in ein Jacobi-System überführt werden können. Dazu werden die folgenden Notationen benötigt. Für $i \geq 1$ bezeichne $H_i(A,b,c) \in \mathbb{R}^{i \times i}$ die i-te partielle Hankelmatrix eines Systems (A,b,c) , d.h.

$$H_i(A,b,c) := \begin{bmatrix} cb & cAb & \dots & cA^{i-1}b \\ cAb & cA^2b & & \vdots \\ \vdots & & & \vdots \\ cA^{i-1}b & \dots & \dots & cA^{2(i-1)}b \end{bmatrix},$$

und es sei

$$\hat{\Sigma}(n,1,1) := \{(A,b,c) \in \Sigma(n,1,1) \mid \det H_i(A,b,c) \neq 0 \text{ für alle } i \in \underline{n}\}.$$

Die Menge $\hat{\Sigma}(n,1,1)$ ist offen und dicht in $\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \times \mathbb{R}^{1 \times n}$ und es gilt:

Satz 1

Zu jedem $(A, b, c) \in \hat{\Sigma}(A, b, c)$ gibt es genau ein $S \in \text{Gl}(n)$ mit

$$(A_J, b_J, c_J) := (S^{-1}AS, S^{-1}b, cS) \in J(n, 1, 1). \quad (16)$$

Falls $(A, b, c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \times \mathbb{R}^{1 \times n}$ und falls es ein $S \in \text{Gl}(n)$ mit (16) gibt, so folgt $(A, b, c) \in \hat{\Sigma}(n, 1, 1)$.

Zum Beweis werden die folgenden Lemmata benötigt.

Lemma 2 (LR-Zerlegung)

Es sei $H \in \mathbb{R}^{n \times n}$. Es gibt Matrizen $L \in \text{Lt}(n)$ und $R \in \text{Ut}^*(n)$ mit $H = L \cdot R$ genau dann, wenn alle führenden Hauptminoren von H ungleich Null sind, d.h. genau dann, wenn

$$\det \begin{bmatrix} h_{11} & \dots & h_{1i} \\ \vdots & & \vdots \\ h_{i1} & \dots & h_{ii} \end{bmatrix} \neq 0 \quad \text{für alle } i \in \underline{n}.$$

Falls $H = L \cdot R$ mit $L \in \text{Lt}(n)$ und $R \in \text{Ut}^*(n)$, so sind L und R eindeutig bestimmt.

Der Beweis von Lemma 2 findet sich in (Wilkinson 1965, Abschn. 4.17-4.20).

Lemma 3

Es seien $H \in \text{Gl}(n)$ und $L, R, \tilde{L}, \tilde{R} \in \mathbb{R}^{n \times n}$ mit $H = L \cdot R = \tilde{L} \cdot \tilde{R}$. Dann existiert eine eindeutig bestimmte Matrix $S \in \text{Gl}(n)$ mit

$$LS = \tilde{L}, \quad S^{-1}R = \tilde{R}. \quad (17)$$

Beweis

Aus $LR = \tilde{L}\tilde{R}$ folgt $(\tilde{L}^{-1}L)^{-1} = L^{-1}\tilde{L} = \tilde{R}\tilde{R}^{-1}$, und daher gilt mit $S := \tilde{R}\tilde{R}^{-1}$:

$$\tilde{R} = \tilde{L}^{-1}LR = S^{-1}R$$

$$\tilde{L} = L\tilde{R}\tilde{R}^{-1} = LS.$$

Die Eindeutigkeit von S ist offensichtlich. □

Beweis von Satz 1

Nach Lemma 2 gibt es eine eindeutig bestimmte Zerlegung $H_n(A,b,c) = LR$ mit $L \in Lt(n)$ und $R \in Ut^*(n)$. Es gilt aber auch $H_n(A,b,c) = O(A,c) \cdot R(A,b)$. Also gibt es nach Lemma 3 eine eindeutig bestimmte Matrix S mit

$$O(A,c)S = L, \quad S^{-1}R(A,b) = R. \quad (18)$$

Die Bedingungen (18) sind aber äquivalent zu

$(S^{-1}AS, S^{-1}b, cS) \in J(n, 1, 1)$. Damit ist die erste Behauptung des Satzes bewiesen. Es sei jetzt $S \in Gl(n)$ so, daß (16) gilt.

Dann folgt (18) und $H_n(A,b,c) = LR$. Aus Lemma 2 folgt dann die Behauptung. □

Das System (A_J, b_J, c_J) aus Satz 1 wird *System-Jacobi-Form* von (A,b,c) genannt.

Bemerkung 16

Der Beweis von Satz 1 liefert im Prinzip eine konstruktive Methode zur Berechnung der System-Jacobi-Form. Es wird dabei aber mit der Erreichbarkeits-, Beobachtbarkeits- und Hankel-

matrix gearbeitet, so daß dieses Vorgehen vom numerischen Gesichtspunkt aus nicht zu empfehlen ist (vgl. Beispiel 1). In Abschnitt 3.2.3. wird ein alternativer Algorithmus vorgestellt, der direkt mit A, b, c arbeitet.

Bemerkung 17

Die Festlegung $a_{i,i-1} = 1$ und $a_{i-1,i} \neq 0$ in Definition 6 ist willkürlich. Beispielsweise hätte man für Jacobi-Systeme genausogut die Festlegung $a_{i,i-1} > 0$, $|a_{i-1,i}| = |a_{i,i-1}|$ für alle $i \in \underline{n}$ treffen können. Dann wäre die System-Jacobi-Form bis auf Vorzeichen symmetrisch.

Das folgende Bild 3 zeigt das Signalflußdiagramm eines Jacobi-Systems

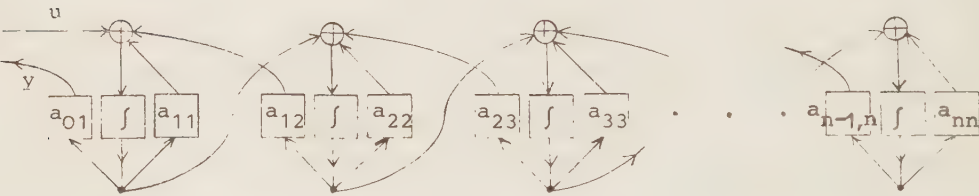


Bild 3

In (Byrnes und Duncan 1982) wird gezeigt, daß sich *symmetrische* Jacobi-Systeme (vgl. Bemerkung 8) aus der Kettenbruchzerlegung einer Transferfunktion vom Cauchy-Index n gewinnen lassen. Dieses legt die Vermutung nahe, daß ein Zusammenhang zwischen der Kettenbruchzerlegung einer rationalen Funktion und den Jacobi-Systemen existiert. Geht man aus von einer rationalen

Funktion $g(s)$, die durch einen Kettenbruch der in (Byrnes und Duncan 1982) betrachteten Form

$$g(s) = \frac{a_{01}}{(s-a_{11}) - \frac{a_{12}}{(s-a_{22}) - \dots - \frac{a_{n-1,n}}{s-a_{nn}}}} \quad (19)$$

repräsentiert ist, und definiert $g_n, \dots, g_1$ rekursiv durch

$$g_n(s) = \frac{1}{s-a_{nn}},$$

$$g_i(s) = \frac{1}{(s-a_{ii}) - a_{i,i+1}g_{i+1}}, \quad i = n-1, \dots, 1,$$

so gilt $g = a_{01}g_1$. Es sei nun $\hat{u}(s)$ die Laplace-Transformierte eines Eingangssignals und $\hat{y}(s) = g(s)\hat{u}(s)$.

Definiere ferner $\hat{x}_1 = g_1\hat{u}$ und $\hat{x}_i = g_i\hat{x}_{i-1}$ für $i = 2, 3, \dots, n$. Dann gilt für $i = 2, 3, \dots, n-1$:

$$\hat{x}_i = \frac{1}{(s-a_{ii}) - a_{i,i+1}g_{i+1}} \hat{x}_{i-1},$$

d.h.

$$\hat{x}_i = \frac{1}{s-a_{ii}} (\hat{x}_{i-1} + a_{i,i+1}\hat{x}_{i+1}). \quad (20)$$

Ferner gilt für $i = n$ und $i = 1$:

$$\hat{x}_n = \frac{1}{s-a_{nn}} \hat{x}_{n-1}, \quad \hat{x}_1 = \frac{1}{s-a_{11}} (\hat{u} + a_{12} \hat{x}_2). \quad (21)$$

Übersetzt man die Gleichungen in den Zeitbereich, so erhält man gerade die Gleichungen eines Jacobi-Systems. Damit ist die eine Richtung der folgenden Proposition bewiesen. Die Umkehrung zeigt man analog.

Proposition 4

Es sei die Kettenbruchzerlegung (19) einer Transferfunktion mit $a_{i-1,i} \neq 0$ für $i \in \underline{n}$ gegeben. Dann ist das Jacobi-System aus Definition 1 eine minimale Realisierung von $g(s)$. Umgekehrt definieren die Koeffizienten eines Jacobi-Systems die Koeffizienten der Kettenbruchzerlegung (19) der Transferfunktion.

Bemerkung 18

Proposition 4 läßt sich in beide Richtungen ausnutzen:

Ist (19) gegeben, so erhält man sofort eine minimale Realisierung von $g(s)$. Andersherum kann man aus der Darstellung

$$g(s) = \frac{c_n + c_{n-1}s + \dots + c_1 s^{n-1}}{a_n + a_{n-1}s + \dots + a_1 s^{n-1} + s^n}$$

einer Transferfunktion die Darstellung (19) berechnen, indem man den Algorithmus zur Herstellung der System-Jacobi-Form (Abschnitt 3.2.3) auf das Hessenberg-System

$$A = \begin{bmatrix} -a_1 & \dots & -a_n \\ 1 & & 0 \\ & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad c = [c_1, \dots, c_n]$$

angewendet.

Bemerkung 19

In der klassischen Theorie der Kettenbrüche (Wall 1967, Abschn. 9.40) werden sogenannte J-Kettenbrüche (J von Jacobi ?) behandelt, die die Form

$$\frac{1}{r_1 s + t_1 + \frac{1}{r_2 s + t_2 + \frac{1}{r_3 s + t_3 + \dots}}} \quad (22)$$

besitzen. Es ist klar, daß die Kettenbrüche (19) und (22) mit einer einfachen Transformation ineinander überführt werden können; sie sind äquivalent im Sinne von (Wall 1967, Abschn. 1.3). Damit läßt sich mit Hilfe von Proposition 4 und Satz 1 eine zu (Wall 1967, Theorem 41.1) alternative Charakterisierung aller strikt proper rationalen Funktionen geben, die eine J-Kettenbruchzerlegung besitzen.

Bemerkung 20

In der Theorie der *Modellreduktion* wurde schon relativ früh vorgeschlagen, die Transferfunktion als Kettenbruch darzustellen und diese Zerlegung dann abzuberechnen (Chen und Shieh 1968). Dort werden jedoch nicht die J-Kettenbrüche, sondern die Kettenbruchzerlegungen vom Stieltjes-Typ (Wall 1967, Abschn. 9.44) für $g(1/s)$ benutzt. Es wird jetzt gezeigt, daß die Kettenbruchzerlegung (19) zu einer minimalen partiellen Realisierung (Kalman 1971) von $g(s)$ führt. Dazu wird die folgende Definition benötigt.

Definition 7

Es sei $(A, b, c) \in J(n, 1, 1)$ und $0 \leq N \leq n$.

Das System $(A_{(N)}, b_{(N)}, c_{(N)}) \in J(N, 1, 1)$, definiert durch
(Bezeichnung wie in Definition 6)

$$A_{(N)} := \begin{bmatrix} a_{11} & a_{12} & & & 0 \\ 1 & & & & \\ & & & & \\ & & & & a_{N-1,N} \\ & & 0 & & 1 & a_{NN} \end{bmatrix}, \quad b_{(N)} := \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad c_{(N)} := [a_{01} 0 \dots 0], \quad (23)$$

heißt das N -te Hauptabschnittssystem von (A, b, c) , und das System $(A^{(N)}, b^{(N)}, c^{(N)}) \in J(n-N, 1, 1)$, definiert durch

$$A^{(N)} := \begin{bmatrix} a_{N+1,N+1} & a_{N+1,N+2} & & & 0 \\ 1 & & & & \\ & & & & \\ & & & & a_{n-1,n} \\ & & 0 & & 1 & a_{n,n} \end{bmatrix}, \quad b^{(N)} := \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (24)$$

$$c^{(N)} := [a_{N,N+1} 0 \dots 0],$$

heißt das N -te Restsystem von (A, b, c) . Hierbei gilt für $N = 0$: $(A^{(N)}, b^{(N)}, c^{(N)}) = (A, b, c)$ und $(A_{(N)}, b_{(N)}, c_{(N)})$ ist nicht vorhanden, und für $N = n$ gilt $(A_{(N)}, b_{(N)}, c_{(N)}) = (A, b, c)$ und $(A^{(N)}, b^{(N)}, c^{(N)})$ ist nicht vorhanden.

Definition 8

Es sei $M = (m_0, m_1, \dots, m_N)^T \in \mathbb{R}^{N+1}$ und

$(A, b, c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \times \mathbb{R}^{1 \times n}$. (A, b, c) heißt *minimale*

partielle Realisierung, falls $cA^i b = m_i$ für alle $i \in \underline{N}$,
und die Dimension n minimal ist unter allen Systemen,
die $cA^i b = m_i$ erfüllen.

Proposition 5

Es sei $(A, b, c) \in J(n, 1, 1)$ und $0 < N \leq n$. Dann ist
 $(A_{(N)}, b_{(N)}, c_{(N)})$ eine minimale partielle Realisierung
von $M = (cb, cAb, \dots, cA^{2(N-1)}b)$.

Beweis

Da die partielle Hankelmatrix $H_N(A, b, c)$ nur von M
abhängt und Rang N hat, hat jede partielle Realisierung
mindestens die Dimension N . Aus der Struktur von (A, b, c)
(vgl. Bild 3) ist aber ersichtlich, daß $cb, cAb, \dots, cA^{2(N-1)}b$
nicht von $a_{N, N+1}$ abhängen, und daher ist $(A_{(N)}, b_{(N)}, c_{(N)})$
eine partielle Realisierung.

□

Bemerkung 21

Proposition 5 ist ein Spezialfall von (Gragg und Lindquist
1983, Theorem 5). Auch der Zusammenhang zwischen partiellen
Realisierungen und Kettenbruchzerlegungen ist bereits bekannt
(Gragg und Lindquist 1983, Abschn. 4). Ferner läßt sich
Satz 1 als Spezialfall von (Bosgra und van der Weiden 1980,
Theorem 3.6 (vgl. Gleichung (3.14) dort)) interpretieren.
Daher enthält dieser Abschnitt keine wesentlich neuen Resul-
tate. Die hier vorgestellte Behandlung des "generischen
Falles" lohnt sich trotzdem, da die Zusammenhänge so trans-
parenter und die Beweise einfacher werden.

3.2. Systeme mit mehreren Eingängen

3.2.1. Hessenberg-Systeme

Im Gegensatz zu Systemen mit einem Eingang gibt es für multivariable Systeme eine Vielzahl von Möglichkeiten, System-Hessenberg-Formen zu definieren. Diese Systeme verdienen nicht immer das Attribut "Hessenberg", da die A-Matrix i.a. keine Hessenberg-Matrix im klassischen Sinn ist. In der folgenden Definition wird der Begriff "Hessenberg-System" jedoch sehr weit gefaßt.

Definition 9

Ein System $(A,B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$ heißt *Hessenberg-System*, falls es paarweise verschiedene Indizes $j_1, j_2, \dots, j_n$ mit $1 \leq j_i \leq m + i - 1$ gibt, so daß für die Einträge $(x_{ij})_{i \in \underline{n}; j \in \underline{n+m}}$ der Matrix $X := [B,A] \in \mathbb{R}^{n \times (n+m)}$ gilt:

$$x_{i,j_i} \neq 0 \quad (25)$$

$$x_{\ell,j_i} = 0 \quad \text{für } \ell > i. \quad (26)$$

Die Einträge x_{i,j_i} werden *Pivotelemente* des Hessenberg-Systems genannt. Die Menge aller Hessenberg-Systeme wird mit $H(n,m)$ bezeichnet.

Bemerkung 22

In einem Hessenberg-System gibt es also in jeder Zeile von X genau ein Pivotelement, welches immer links von der Diagonalen in A liegt. Ein Pivotelement ist stets ungleich

Null und die Einträge unterhalb eines Pivotelementes verschwinden.

Beispiel 4

Die Struktur eines Hessenberg-Systems mit $n = 9$, $m = 5$,
 $j_1 = 2$, $j_2 = 5$, $j_3 = 6$, $j_4 = 1$, $j_5 = 8$, $j_6 = 4$, $j_7 = 3$,
 $j_8 = 11$, $j_9 = 9$ ist gegeben durch

$$B = \begin{bmatrix} * & * & * & * & * \\ * & 0 & * & * & * \\ * & 0 & * & * & 0 \\ * & 0 & * & * & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * & * & * \\ 0 & * & 0 & * & * & * & * & * & * \\ 0 & * & 0 & * & * & * & * & * & * \\ 0 & * & 0 & * & * & * & * & * & * \\ 0 & * & 0 & * & * & * & * & * & * \\ 0 & * & 0 & * & * & 0 & * & * & * \end{bmatrix}.$$

Bemerkung 23

Definition 1 und 9 sind konsistent, und ein Hessenberg-System ist stets erreichbar (Hautus Test!).

Der folgende Algorithmus SHFM (System Hessenberg Form; multivariabel) transformiert ein beliebiges System $(A, B) \in \Sigma(n, m)$ in ein orthogonal ähnliches Hessenberg-System. Da Systeme häufig eine Ausgangsmatrix $C \in \mathbb{R}^{p \times n}$ besitzen, wird diese im Algorithmus mitgeführt. Sie hat jedoch keinen Einfluß auf den Algorithmus. Man beachte, daß der Algorithmus für $C = I \in \mathbb{R}^{n \times n}$ die Transformationsmatrix $Q \in O(n)$ liefert.

8. Wähle j_k ($1 \leq j_k \leq m + k - 1$) so, daß die j_k -te Spalte $(\bar{X}^k)_{j_k}$ von $\bar{X}^k$ von Null verschieden ist.

9. Wähle $\bar{Q}^k \in O(n-k+1)$ so, daß

$$(\bar{Q}^k)^T (\bar{X}^k)_{j_k}$$

ein Vielfaches des ersten Einheitsvektors ist.

10. Berechne

$$X^{k+1} = \begin{bmatrix} I & O \\ O & \bar{Q}^k \end{bmatrix}^T X^k \begin{bmatrix} I & O \\ O & \bar{Q}^k \end{bmatrix}.$$

11. $k \leftarrow k+1$.

12. Gehe zu 4.

13. Bestimme $\tilde{A}_{11}, \tilde{A}_{12}, \tilde{A}_{22}, \tilde{C}_1, \tilde{C}_2, \tilde{B}_1$,
definiert durch

$$X^k = \begin{bmatrix} O & \tilde{C}_1 & \tilde{C}_2 \\ \tilde{B}_1 & \tilde{A}_{11} & \tilde{A}_{12} \\ O & O & \tilde{A}_{22} \end{bmatrix} \begin{matrix} \} p \\ \} k-1 \\ \} n-k+1 \end{matrix} = \begin{bmatrix} O & \tilde{C} \\ \tilde{B} & \tilde{A} \end{bmatrix} \begin{matrix} \} p \\ \} m \end{matrix}$$

(falls $p, k-1$ oder $n-k+1$ verschwindet, so treten entsprechende Zeilen und Spalten nicht auf).

14. Kommentar: Es gilt

$$\tilde{A} = \tilde{Q}^T A \tilde{Q}$$

$$\tilde{B} = \tilde{Q}^T B$$

$$\tilde{C} = C \tilde{Q}$$

für eine Matrix $\tilde{Q} \in O(n)$. Ferner gilt

$$(\tilde{A}_{11}, \tilde{B}_1) \in H(k-1, m)$$

und (A, B) ist kontrollierbar genau dann, wenn $k-1 = n$.

15. Ausgabe $\tilde{A}, \tilde{B}, \tilde{C}, k-1, j_1, \dots, j_{k-1}$.

16. Stoppe.

Im Schritt 8 des Algorithmus SHFM besteht noch eine gewisse Freiheit in der Wahl von j_k . Legt man a priori eine Reihenfolge fest, in der die Spalten von $\bar{X}^k$ überprüft werden sollen, und nimmt man die erste auftretende von Null verschiedene Spalte, so erhält man eine Normalform unter orthogonaler Ähnlichkeit, vorausgesetzt, daß man zusätzlich noch das Vorzeichen der Pivotelemente festlegt. Um dieses zu präzisieren, seien n (totale) Ordnungen $<_1, <_2, \dots, <_n$ auf $\underline{m}$ gegeben. Ferner seien $i_1^k, i_2^k, \dots, i_m^k \in \underline{m}$ definiert durch $i_1^k <_k i_2^k <_k \dots <_k i_m^k$. Zu einem Hessenberg-System $(A, B) \in H(n, m)$ mit Pivotelementen $x_{1, j_1}, x_{2, j_2}, \dots, x_{n, j_n}$ in $X := [B, A]$ sei für $k \in \underline{n}$ die Matrix $\bar{X}_k \in R^{(n-k+1) \times (m+k-1)}$ definiert durch

$$X = \begin{pmatrix} * & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{\bar{X}_k}_{m+k-1} & \underbrace{*}_{n-k+1} & \dots & * \end{pmatrix}_{k-1}^{n-k+1} \quad (27)$$

In der Matrix $\bar{X}_k$ verschwinden die Spalten $j_1, j_2, \dots, j_{k-1}$. Nach Streichung dieser Spalten erhält man eine Matrix der Form

$$\tilde{X}_k = [z_k^1, z_k^2, \dots, z_k^m] \in \mathbb{R}^{(n-k+1) \times m} \quad (28)$$

In der Matrix $\tilde{X}_k$ gibt es eine Spalte (sagen wir $z_k^{\ell(k)}$, $1 \leq \ell(k) \leq m$) mit $z_k^{\ell(k)} = [\gamma \ 0 \dots 0]^T$, wobei γ das k -te Pivotelement von (A, B) ist.

Definition 10

Ein System $(A, B) \in H(n, m)$ heißt $(\prec_1, \prec_2, \dots, \prec_n)$ -Hessenberg-System, falls für alle $k \in \underline{n}$ die Spalte $z_k^{\ell(k)}$ in der Ordnung $\prec_k$ die erste nicht verschwindende Spalte von $\tilde{X}_k$ ist, und zusätzlich alle Pivotelemente positiv sind.

Mit diesen Vorbereitungen kann man den folgenden Satz beweisen.

Satz 3

Es seien $\prec_1, \prec_2, \dots, \prec_n$ gegeben. Zu jedem $(A, B) \in \Sigma(n, m)$ gibt es genau ein $Q \in O(n)$, so daß $(Q^T A Q, Q^T B)$ ein $(\prec_1, \prec_2, \dots, \prec_n)$ -Hessenberg-System ist.

Die Existenzaussage folgt aus dem Algorithmus SHFM:

Beweis (Existenz)

Streicht man in der Matrix $\bar{X}^k$ in Schritt 6 des Algorithmus die Spalten $j_1, \dots, j_{k-1}$, und wählt in der entstehenden Matrix $\hat{X}^k \in \mathbb{R}^{(n-k+1) \times m}$ die in der Ordnung $<_k$ erste von Null verschiedene Spalte als $(\bar{X}^k)_{j_k}$ in Schritt 8, so erhält man ein $(<_1, \dots, <_n)$ -Hessenberg System, wenn man die Pivotelemente positiv wählt.

□

Um die Eindeutigkeitsaussage in Satz 3 zu zeigen, sind noch einige Vorbereitungen nötig. Dazu seien für $k \in \underline{n}$ und $x \in S^{k-1} := \{x \in \mathbb{R}^k \mid x^T x = 1\}$ die Householdermatrizen (Stoer und Bulirsch 1976, Abschn. 4.7) $P_k(x) \in O(n)$ definiert durch

$$P_k(x) := \begin{cases} I & \text{falls } x = e_1 \\ \begin{bmatrix} I & \tilde{0} \\ 0 & I - 2 \frac{(x-e_1)(x-e_1)^T}{(x-e_1)^T(x-e_1)} \end{bmatrix} & \text{falls } x \neq e_1 \end{cases}$$

wobei $e_1 := [1 \ 0 \dots 0]^T \in \mathbb{R}^k$. Man rechnet schnell nach, daß die $(n-k+1)$ -te Spalte von $P_k(x)$ gegeben ist durch $\tilde{x} := \begin{bmatrix} 0 \\ x \end{bmatrix}$. Mit Hilfe dieser Matrizen lassen sich die orthogonalen Matrizen parametrisieren:

Lemma 5

Zu jedem $Q \in O(n)$ gibt es eindeutig bestimmte Vektoren $x_i \in S^{i-1}$, $i \in \underline{n}$, mit

$$Q = P_n(x_n) \cdot P_{n-1}(x_{n-1}) \cdot \dots \cdot P_2(x_2) \cdot P_1(x_1). \quad (29)$$

Beweis

Existenz: Jede invertierbare Matrix läßt sich mit Hilfe von Householdermatrizen in eine obere Dreiecksmatrix mit positiven Diagonalelementen transformieren (Stoer und Bulirsch 1976, Abschn. 4.7). Ist die Ausgangsmatrix orthogonal, so haben die Householdermatrizen die Form $P_i(x_i)$, und die obere Dreiecksmatrix ist die Einheitsmatrix.

Eindeutigkeit: Die erste Spalte der Matrix auf der rechten Seite von (29) ist x_n . Daher ist x_n durch Q eindeutig bestimmt. Nimmt man jetzt an, daß $x_n, x_{n-1}, \dots, x_k$ für $k > 1$ eindeutig bestimmt sind, so ist auch die Matrix

$$P_k(x_k) \cdot \dots \cdot P_n(x_n) \cdot Q = P_{k-1}(x_{k-1}) \cdot \dots \cdot P_1(x_1) \quad (30)$$

eindeutig bestimmt. Da die $(n-k+2)$ -te Spalte der rechten Seite von (30) durch x_{k-1} gegeben ist, folgt die Eindeutigkeit von x_{k-1} .

□

Beweis der Eindeutigkeitsaussage in Satz 3

Es seien $Q \in O(n)$ und $\hat{Q} \in O(n)$, so daß $(Q^T A Q, Q^T B)$ und $(\hat{Q}^T A \hat{Q}, \hat{Q}^T B)$ jeweils $(<_1, \dots, <_n)$ -Hessenberg-Systeme sind. Die Annahme $Q \neq \hat{Q}$ wird zum Widerspruch geführt. Nach Lemma 5 gibt es $x_i, \hat{x}_i \in S^{i-1}$, $i \in \underline{n}$, mit

$$\begin{aligned} Q &= P_n(x_n) \cdot \dots \cdot P_2(x_2) \cdot P_1(x_1), \\ \hat{Q} &= P_n(\hat{x}_n) \cdot \dots \cdot P_2(\hat{x}_2) \cdot P_1(\hat{x}_1). \end{aligned}$$

Es sei $k := \min\{i \in \underline{n} \mid x_{n-i+1} \neq \hat{x}_{n-i+1}\}$, und die Matrizen $Y, X, \hat{X} \in \mathbb{R}^{n \times (n+m)}$ seien definiert durch

$$Y = P_{n-k+2}(x_{n-k+2}) \cdot \dots \cdot P_n(x_n) \cdot [B, A] \begin{bmatrix} I & O \\ O & P_n(x_n) \cdot \dots \cdot P_{n-k+2}(x_{n-k+2}) \end{bmatrix},$$

$$X = Q^T[B, A] \begin{bmatrix} I & O \\ O & Q \end{bmatrix},$$

$$\hat{X} = \hat{Q}^T[B, A] \begin{bmatrix} I & O \\ O & \hat{Q} \end{bmatrix}.$$

Dann gilt

$$X = P_1(x_1) \cdot \dots \cdot P_{n-k+1}(x_{n-k+1}) \cdot Y \cdot \begin{bmatrix} I & O \\ O & P_{n-k+1}(x_{n-k+1}) \cdot \dots \cdot P_1(x_1) \end{bmatrix};$$

$$\hat{X} = P_1(\hat{x}_1) \cdot \dots \cdot P_{n-k+1}(\hat{x}_{n-k+1}) \cdot Y \cdot \begin{bmatrix} I & O \\ O & P_{n-k+1}(\hat{x}_{n-k+1}) \cdot \dots \cdot P_1(\hat{x}_1) \end{bmatrix}.$$

Die Matrizen Y , X und $\hat{X}$ seien wie folgt partitioniert:

$$Y = \begin{bmatrix} * & * \\ \bar{Y} & * \end{bmatrix}_{n-k+1}, \quad X = \begin{bmatrix} * & * \\ \bar{X}_k & * \end{bmatrix}_{n-k+1}, \quad \hat{X} = \begin{bmatrix} * & * \\ \hat{\bar{X}}_k & * \end{bmatrix}_{n-k+1}.$$

Dann hat sowohl $\bar{X}_k$ als auch $\hat{\bar{X}}_k$ in der ersten Zeile genau ein Pivotelement. Ferner folgt aus den Gleichungen

$$P_1(x_1) \cdot \dots \cdot P_{n-k+1}(x_{n-k+1}) = \begin{bmatrix} I & O \\ O & R \end{bmatrix}_{n-k+1}; \quad (30a)$$

$$P_1(\hat{x}_1) \cdot \dots \cdot P_{n-k+1}(\hat{x}_{n-k+1}) = \begin{bmatrix} I & O \\ O & \hat{R} \end{bmatrix}_{n-k+1},$$

die Identität

$$\bar{X}_k = R\bar{Y}; \quad \hat{\bar{X}}_k = \hat{R}\bar{Y}. \quad (31)$$

Also haben die Spalten von $\bar{X}_k$ und $\hat{\bar{X}}_k$ zumindest gleiche Norm. Aufgrund der Definition eines $(\langle_1, \dots, \langle_n)$ -Hessenberg-Systems folgt hieraus aber, daß die Pivotelemente in der ersten Zeile von $\bar{X}_k$ und $\hat{\bar{X}}_k$ jeweils in der gleichen Spalte stehen und gleichen Betrag haben. Aus der entsprechenden Spalte in (31) ergibt sich dann, daß die erste Spalte von R^{-1} mit der ersten Spalte von $\hat{R}^{-1}$ übereinstimmt. Aus (30a) folgt aber, daß diese Spalten gerade durch x_{n-k+1} und $\hat{x}_{n-k-1}$ gegeben sind. Die Identität $x_{n-k+1} = \hat{x}_{n-k+1}$ steht aber im Widerspruch zur Definition von k .

□

Das System $(Q^T A Q, Q^T B)$ aus Satz 3 heißt $(\langle_1, \dots, \langle_n)$ -Hessenberg-Form von (A, B) .

Definition 11

- (i) Falls die Ordnungen $\langle_1, \dots, \langle_n$ auf $\underline{m}$ für alle $i \in \underline{n}$ definiert sind durch $m \langle_i m-1 \langle_i \dots \langle_i 2 \langle_i 1$, so heißt die $(\langle_1, \dots, \langle_n)$ -Hessenberg-Form auch *Hermite-Hessenberg-Form*.
- (ii) Falls die Ordnungen $\langle_1, \dots, \langle_n$ auf $\underline{m}$ für alle $i \in \underline{n}$ definiert sind durch $1 \langle_i 2 \langle_i \dots \langle_i m-1 \langle_i m$, so heißt die $(\langle_1, \dots, \langle_n)$ -Hessenberg Form auch *Kronecker-Hessenberg-Form*.

Bemerkung 24

Für $n = 8$ und $m = 4$ ist die Struktur einer typischen Hermite-Hessenberg-Form gegeben durch

$$B = \begin{bmatrix} * & * & * & \textcircled{*} \\ * & * & * & 0 \\ * & * & * & 0 \\ * & \textcircled{*} & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ \textcircled{*} & 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} * & * & * & * & * & * & * & * \\ \textcircled{*} & * & * & * & * & * & * & * \\ 0 & \textcircled{*} & * & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * & * \\ 0 & 0 & 0 & \textcircled{*} & * & * & * & * \\ 0 & 0 & 0 & 0 & \textcircled{*} & * & * & * \\ 0 & 0 & 0 & 0 & 0 & \textcircled{*} & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}.$$

Die Existenz der Hermite-Hessenberg-Form (teilweise in leicht abgewandelter Form) wurde bereits in (Aplavich 1974; Nour Eldin 1974) gezeigt. Die Eindeutigkeitsaussage findet sich in (Hinrichsen und Linnemann 1982). Die Struktur einer Hermite-Hessenberg-Form läßt sich auch durch die sog. *Hermite-Indizes* (Hinrichsen und Prätzel-Wolters 1983) beschreiben. Die Hermite-Hessenberg-Form hat den wesentlichen Vorteil, daß sich für *strukturierte Systeme* (Lin 1974) die *generischen Hermite-Indizes* berechnen lassen (Hinrichsen und Linnemann 1984). Mit Hilfe dieser Indizes können durch Rundungsfehler erzeugte, falsche Ergebnisse des Algorithmus SHFM dann teilweise erkannt werden.

Bemerkung 25

Für $n = 8$ und $m = 4$ ist die Struktur einer typischen Kronecker-Hessenberg-Form gegeben durch

$$B = \begin{bmatrix} \odot & * & * & * \\ 0 & \odot & * & * \\ 0 & 0 & 0 & \odot \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad A = \begin{bmatrix} * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ \odot & * & * & * & * & * & * & * \\ 0 & \odot & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & \odot & * & * & * \\ 0 & 0 & 0 & 0 & 0 & \odot & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & \odot & * \end{bmatrix}.$$

Die Existenz der Kronecker-Hessenberg-Form (teilweise in leicht abgewandelter Form) wurde bereits in (Munro und McLeod 1971; Tse, Medanic und Perkins 1978; Nour Eldin und Heister 1980; Konstantinov, Petkov und Christov 1980; Paige 1981; Patel 1981; van Dooren 1981; Varga 1981) gezeigt. Die Eindeutigkeitsaussage findet sich in (Konstantinov, Petkov und Christov 1981). Die Struktur der Kronecker-Hessenberg-Form läßt sich auch durch die sog. *Kronecker-Indizes* beschreiben.

Bemerkung 26

Die Erreichbarkeitsmatrix einer Hermite- oder Kronecker-Hessenberg-Form enthält, an einer durch die Hermite- bzw. Kronecker-Indizes festgelegten Stelle, eine obere Dreiecksmatrix mit positiven Diagonalelementen. Dadurch läßt sich die Eindeutigkeit dieser Formen mit Hilfe der Eindeutigkeit der QR-Zerlegung beweisen (Konstantinov, Petkov und Christov 1981; Hinrichsen und Linnemann 1984). Wie das folgende Beispiel zeigt, enthält die Erreichbarkeitsmatrix einer $(\langle_1, \dots, \langle_n)$ -Hessenberg Form i.a. keine obere Dreiecksmatrix.

Beispiel 5

Es sei $n = 4$, $m = 2$ und $1 <_1 2$; $1 <_2 2$; $2 <_3 1$;
 $2 <_4 1$. Dann ist

$$B = \begin{bmatrix} \textcircled{1} & 1 \\ 0 & \textcircled{1} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & \textcircled{1} & 1 & 1 \\ 1 & 0 & \textcircled{1} & 1 \end{bmatrix}$$

eine $(<_1, \dots, <_4)$ -Hessenberg Form. Die Erreichbarkeitsmatrix $R(A, B)$ enthält jedoch keine obere Dreiecksmatrix.

Bemerkung 27

Das Beispiel 5 zeigt insbesondere, daß sich nicht jeder Ansammlung $(<_1, \dots, <_n)$ von n Ordnungen $<_1, \dots, <_n$ eine "nice order" im Sinne von *Hinrichsen* und *Prätzel-Wolters* (1983) derart zuordnen läßt, daß für alle $(<_1, \dots, <_n)$ -Hessenberg-Formen (A, B) die durch das entsprechende Streichverfahren gebildete $(n \times n)$ -Teilmatrix von $R(A, B)$ eine obere Dreiecksmatrix ist. Umgekehrt läßt sich auch zeigen, daß sich nicht jeder "nice order" eine Ansammlung $(<_1, \dots, <_n)$ von n Ordnungen derart zuordnen läßt, daß alle Systeme (A, B) , für die das Streichverfahren eine obere Dreiecksmatrix liefert, $(<_1, \dots, <_n)$ -Hessenberg-Systeme sind. Die Konzepte " $(<_1, \dots, <_n)$ -Hessenberg-Form" und "nice order" sind also nicht direkt vergleichbar.

Bemerkung 28

Die in Abschnitt 3.1.1. gemachten Bemerkungen über numerische Stabilität und Kondition treffen auch hier zu. Insbesondere läßt sich der Algorithmus SHFM numerisch stabil implementieren, und die $(\langle_1, \dots, \langle_n)$ -Hessenberg-Form kann beliebig schlecht konditioniert sein (die Systeme in den Beispielen 2 und 3 sind auch $(\langle_1, \dots, \langle_n)$ -Hessenberg-Formen !). Zusätzlich tritt hier jedoch noch das Problem auf, daß die Abbildung, die jedem erreichbaren System die $(\langle_1, \dots, \langle_n)$ -Hessenberg-Form zuordnet, unstetig ist. Damit zusammenhängende Probleme werden in (Hinrichsen und Linnemann 1984) für die Hermite-Hessenberg-Form vorgestellt. Ist man jedoch nicht an einer Normalform interessiert und will lediglich ein orthogonal ähnliches Hessenberg-System berechnen, so kann es empfehlenswert sein, sich nicht a priori auf Ordnungen $\langle_1, \dots, \langle_n$ festzulegen, sondern sich vielmehr während der Durchführung des Algorithmus für gewisse Pivotelemente zu entscheiden.

Bemerkung 29

Bei Hermite-Hessenberg-Systemen ist die A-Matrix eine Hessenberg-Matrix und bei Kronecker-Hessenberg-Systemen ist die A-Matrix eine Block-Hessenberg-Matrix. Im allgemeinen besitzt ein Hessenberg-System jedoch keine Hessenberg-Struktur im klassischen Sinn. Man beachte jedoch, daß sich jedes Hessenberg-System so permutieren läßt,

daß gekoppelte Hessenberg-Systeme mit einem Eingang entstehen. So läßt sich das System aus Beispiel 4 in das folgende System permutieren:

$$B = \begin{bmatrix} \textcircled{x} & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & \textcircled{x} & x & x & x \\ 0 & 0 & \textcircled{x} & x & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \textcircled{x} & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{x} \end{bmatrix}; \quad A = \begin{bmatrix} x & x & x & x & x & x & x & x & x \\ \textcircled{x} & x & x & x & x & x & x & x & x \\ 0 & \textcircled{x} & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x & x \\ 0 & 0 & x & x & \textcircled{x} & x & 0 & x & x \\ 0 & 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & \textcircled{x} & x & x \\ 0 & 0 & x & x & x & x & x & x & x \end{bmatrix}.$$

3.2.2 Newton-Systeme

In diesem Abschnitt werden die Resultate nicht in ihrer vollen Allgemeinheit dargestellt. Vielmehr wird anhand von Beispiel 4 angedeutet, wie sich ein Hessenberg-System durch Anwendung von nicht-orthogonalen Ähnlichkeitstransformationen noch weiter vereinfachen läßt.

In (Aplevich 1974) - vgl. auch (Nour Eldin und Heister 1980; Hinrichsen und Prätzel-Wolters 1984) - wurde gezeigt, daß sich Hermite- und Kronecker-Hessenberg-Systeme in die sogenannten "Luenberger kanonischen Formen" transformieren lassen. Dabei können die Parameter entweder in den Zeilen oder in den Spalten stehen. Für allgemeinere Hessenberg-Systeme ist dieses nicht mehr möglich, da die Struktur eines Hessenberg-Systems i.a. nicht invariant gegenüber Ähnlichkeitstransformationen $S \in U_t(n)$ ist. Der Algorithmus SHFM läßt sich jedoch so modifizieren, daß auch die Parameter oberhalb der Pivotelemente ausgeräumt werden. Ein System mit den gleichen Indizes wie in Beispiel 4 hätte dann die Form

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad A = \begin{bmatrix} 0 & * & 0 & 0 & * & 0 & * & * & * \\ 0 & * & 0 & 0 & * & 0 & * & * & * \\ 1 & * & 0 & 0 & * & 0 & * & * & * \\ 0 & * & 0 & 0 & * & 0 & * & * & * \\ 0 & * & 1 & 0 & * & 0 & * & * & * \\ 0 & * & 0 & 0 & * & 0 & * & * & * \\ 0 & * & 0 & 0 & * & 0 & * & * & * \\ 0 & * & 0 & 0 & * & 1 & * & * & * \\ 0 & * & 0 & 1 & * & 0 & * & * & * \end{bmatrix}. \quad (35)$$

Durch Permutation entsteht hieraus das System

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; A = \begin{bmatrix} 0 & 0 & * & * & 0 & * & 0 & * & * \\ 1 & 0 & * & * & 0 & * & 0 & * & * \\ 0 & 1 & * & * & 0 & * & 0 & * & * \\ 0 & 0 & * & * & 0 & * & 0 & * & * \\ 0 & 0 & * & * & 0 & * & 0 & * & * \\ 0 & 0 & * & * & 1 & * & 0 & * & * \\ 0 & 0 & * & * & 0 & * & 0 & * & * \\ 0 & 0 & * & * & 0 & * & 1 & * & * \\ 0 & 0 & * & * & 0 & * & 0 & * & * \end{bmatrix}. \quad (36)$$

Diese Form entspricht der "first canonical form" von Luenberger (1967) und ist eine Normalform, falls man im modifizierten Algorithmus SHFM die Ordnungen $\langle_1, \dots, \langle_n$ à priori festlegt.

Durch Ähnlichkeitstransformationen $S \in \text{Ut}(n)$ und Zustandsrückführung läßt sich jedes Hessenberg-System so verändern, daß auf der Diagonalen, der unteren Nebendiagonalen und oberhalb der Diagonalen von A beliebige Werte erscheinen. Dieses geschieht durch einen zu Algorithmus 1 ähnlichen Algorithmus. Die Rückführungsoperationen sind erforderlich, wenn ein Pivotelement "in B steht". Für das System aus Beispiel 4 gibt es also beispielsweise $R \in \text{Ut}(n)$ und $F \in \mathbb{R}^{m \times n}$ mit

$$R^{-1}B = \begin{bmatrix} * & \odot & * & * & * \\ * & 0 & * & * & \odot \\ * & 0 & * & * & 0 \\ \odot & 0 & * & * & 0 \\ 0 & 0 & * & * & 0 \\ 0 & 0 & * & \odot & 0 \\ 0 & 0 & \odot & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad R^{-1}(A+BF)R = \begin{bmatrix} s_1^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & s_2^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \odot & * & s_3^0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & s_4^0 & 0 & 0 & 0 & 0 \\ 0 & * & * & * & s_5^0 & 0 & 0 & 0 \\ 0 & * & 0 & * & * & s_6^0 & 0 & 0 \\ 0 & * & 0 & * & * & * & s_7^0 & 0 \\ 0 & * & 0 & * & * & \odot & * & s_8^0 \\ 0 & * & 0 & \odot & * & 0 & * & s_9^0 \end{bmatrix}. \quad (37)$$

mit $c_{ii} \neq 0$ für $i \in \underline{p}$ und $a_{i,i+p} \neq 0$ für $i \in \underline{n-p}$.

Die Menge aller Jacobi-Systeme wird mit $J(n,m,p)$ bezeichnet.

Es wird nun untersucht, welche Systeme sich in ein Jacobi-System transformieren lassen. Dazu seien die Einträge der Hankel-Matrix $H(A,B,C) := O(A,C) \cdot R(A,B)$ mit h_{ij} bezeichnet, und die i -te partielle Hankel-Matrix sei definiert durch

$$H_i(A,B,C) := \begin{bmatrix} h_{11} & \dots & h_{1i} \\ \vdots & & \vdots \\ h_{i1} & \dots & h_{ii} \end{bmatrix}$$

Dann ist die Menge

$$\hat{\Sigma}(n,m,p) := \{(A,B,C) \in \Sigma(n,m,p) \mid \det H_i(A,B,C) \neq 0 \text{ für alle } i \in \underline{n}\}$$

offen und dicht in $\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n}$, und es gilt die folgende Verallgemeinerung von Satz 1.

Satz 4

Zu jedem $(A,B,C) \in \hat{\Sigma}(n,m,p)$ gibt es genau ein $S \in GL(n)$ mit

$$(A_J, B_J, C_J) := (S^{-1}AS, S^{-1}B, CS) \in J(n,m,p). \quad (37)$$

Falls $(A,B,C) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n}$ und es ein S mit (37) gibt, so folgt $(A,B,C) \in \hat{\Sigma}(n,m,p)$.

Zum Beweis werden die folgenden Verallgemeinerungen der Lemmata 2 und 3 benötigt.

Lemma 6

Es sei $H \in \mathbb{R}^{k \times \ell}$ mit $\text{rg } H = n$. Es gibt Matrizen $L \in \mathbb{R}^{k \times n}$ und $R \in \mathbb{R}^{n \times \ell}$ der Form

$$L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}; \quad L_1 \in \text{Lt}(n); \quad R = [R_1 | R_2]; \quad R_1 \in \text{Ut}^*(n) \quad (38)$$

mit $H = LR$ genau dann, wenn alle führenden Hauptminoren von H ungleich Null sind, d.h. genau dann, wenn

$$\det \begin{bmatrix} h_{11} & \dots & h_{1i} \\ \vdots & & \vdots \\ h_{i1} & \dots & h_{ii} \end{bmatrix} \neq 0 \quad \text{für alle } i \in \underline{n}.$$

Falls $H = LR$ mit L und R wie in (38), so sind L und R eindeutig bestimmt.

Beweis

Falls L und R die Form (38) besitzen und $H = LR$, so sind die ersten n führenden Hauptminoren von H ungleich Null. Es seien nun die ersten n führenden Hauptminoren von H ungleich Null, und H sei wie folgt partitioniert:

$$H = \begin{bmatrix} \underbrace{H_{11} \quad H_{12}}_{n} \quad \underbrace{\quad}_{\ell-n} \\ \underbrace{H_{21} \quad H_{22}}_{n} \quad \underbrace{\quad}_{\ell-n} \end{bmatrix}.$$

Nach Lemma 2 gibt es $L_1 \in \text{Lt}(n)$ und $R_1 \in \text{Ut}^*(n)$ mit $H_{11} = L_1 R_1$. Da $\text{rg } H = n$, folgt somit

$$H \begin{bmatrix} R_1^{-1} & -H_{11}^{-1}H_{12} \\ 0 & I \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ H_{21}R_1^{-1} & 0 \end{bmatrix},$$

d.h.

$$H = \begin{bmatrix} L_1 & 0 \\ H_{21}R_1^{-1} & 0 \end{bmatrix} \begin{bmatrix} R_1 & R_1H_{11}^{-1}H_{12} \\ 0 & I \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} [R_1 \ R_2]$$

für $L_2 := H_{21}R_1^{-1}$ und $R_2 := R_1H_{11}^{-1}H_{12}$. Um die Eindeutigkeit zu zeigen, sei $H = LR = \bar{L} \bar{R}$, wobei $L; \bar{L}$ und $R; \bar{R}$ jeweils die Form (38) haben. Aus Lemma 2 folgt $L_1 = \bar{L}_1$ und $R_1 = \bar{R}_1$, und wegen der Gleichungen $L_1R_2 = \bar{L}_1\bar{R}_2$; $L_2R_1 = \bar{L}_2\bar{R}_1$ folgt auch $R_2 = \bar{R}_2$ und $L_2 = \bar{L}_2$. □

Lemma 7

Es seien $H \in \mathbb{R}^{k \times \ell}$; $L, \tilde{L} \in \mathbb{R}^{k \times n}$; $R, \tilde{R} \in \mathbb{R}^{n \times \ell}$ mit $\text{rg } H = n$ und $H = LR = \tilde{L} \tilde{R}$. Dann gibt es eine eindeutig bestimmte Matrix $S \in \text{Gl}(n)$ mit $LS = \tilde{L}$; $S^{-1}R = \tilde{R}$.

Beweis

Man wähle $T_1 \in \text{Gl}(k)$ und $T_2 \in \text{Gl}(\ell)$ so, daß

$$T_1 H T_2 = \begin{bmatrix} H_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

mit $H_{11} \in \text{Gl}(n)$, und partitioniere die Faktoren entsprechend:

$$T_1 L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}; \quad T_1 \tilde{L} = \begin{bmatrix} \tilde{L}_1 \\ \tilde{L}_2 \end{bmatrix}, \quad R T_2 = [R_1 \ R_2], \quad \tilde{R} T_2 = [\tilde{R}_1 \ \tilde{R}_2].$$

Dann folgt $L_2 = \tilde{L}_2 = 0$; $R_2 = \tilde{R}_2 = 0$, und die Behauptung ergibt sich aus Lemma 3.

□

Beweis von Satz 4

Nach den Lemmata 6 und 7 gibt es eine eindeutig bestimmte Matrix $S \in \text{Gl}(n)$, so daß

$$L := O(A, C)S = O(S^{-1}AS, CS);$$

$$R := S^{-1}R(A, B) = R(S^{-1}AS, S^{-1}B)$$

die Form (38) besitzen. Dieses ist aber äquivalent zu $(S^{-1}AS, S^{-1}B, CS) \in J(n, m, p)$. Andererseits, falls $(S^{-1}AS, S^{-1}B, CS) \in J(n, m, p)$, so gilt $H(A, B, C) = LR$, wobei $L; R$ die Form (38) besitzen. Die Behauptung folgt dann aus Lemma 6.

□

Das System (A_J, B_J, C_J) aus Satz 4 wird System-Jacobi-Form von (A, B, C) genannt.

Bemerkung 30

Der Graph (vereinfachtes Signalflußdiagramm) eines Jacobi-Systems für $n = 6$ und $m = p = 2$ hat die in Bild 4 dargestellte Form



Bild 4

Auf einen eventuellen Zusammenhang zwischen multivariablen System-Jacobi-Formen und Kettenbruchzerlegungen bzw. partiellen Realisierungen (vgl. Abschn. 3.1.3.) wird hier nicht eingegangen. Dafür werden aber noch zwei weitere angenehme Eigenschaften von Jacobi-Systemen vorgestellt.

Das folgende Stabilisierbarkeitsresultat impliziert, daß die Kenntnis eines Hauptabschnittssystems eines Jacobi-Systems (Definition 7 läßt sich einfach auf den multivariablen Fall übertragen) genügt, um das Gesamtsystem zu stabilisieren, vorausgesetzt, daß das dazugehörige Restsystem bereits stabil ist.

Satz 5

Es sei $(A_1, B_1, C_1) \in J(N, m, p)$. Dann gibt es für alle $\varepsilon > 0$ einen dynamischen Regler für (A_1, B_1, C_1) der Form

$$\begin{aligned} u(t) &= M_\varepsilon w(t) + N_\varepsilon y(t) \\ \dot{w}(t) &= F_\varepsilon w(t) + E_\varepsilon y(t), \end{aligned} \quad (39)$$

wobei $w(t) \in \mathbb{R}^s$, welcher (A_1, B_1, C_1) stabilisiert und die folgende zusätzliche Eigenschaft besitzt:

Für alle asymptotisch stabilen Systeme

$(A_2, B_2, C_2) \in \mathbb{R}^{\ell \times \ell} \times \mathbb{R}^{\ell \times m} \times \mathbb{R}^{p \times \ell}$ ($\ell \geq 0$) gibt es ein $\varepsilon_0 > 0$, so daß für $0 < \varepsilon \leq \varepsilon_0$ der Regler (39) auch das zusammengesetzte System

$$A: = \begin{bmatrix} A_1 & 0 \\ \hline OB_2 & A_2 \end{bmatrix}; \quad B: = \begin{bmatrix} B_1 \\ \hline 0 \end{bmatrix}; \quad C: = [C_1 | 0] \quad (40)$$

stabilisiert.

Beweis

Der Beweis ergibt sich unmittelbar aus (Linnemann 1984, Lemma 1) und (Willems und Ikeda 1984, Lemma 5), wenn man bedenkt, daß $V_{b, \ker[0 \ B_2]}^* = \mathbb{R}^n$ und $S_{b, \text{im} \begin{bmatrix} 0 \\ C_2 \end{bmatrix}}^* = \{0\}$ für das Jacobi-System (A_1, B_1, C_1) gilt (vgl. auch Kapitel 5).

□

Satz 5 mit $m = p = 1$ zusammen mit Proposition 5 kann man so interpretieren, daß sich das Gesamtsystem mit Hilfe des reduzierten Modells (Hauptabschnittssystems; vgl. Def. 7) stabilisieren läßt, vorausgesetzt der vernachlässigte Teil (Restsystem; vgl. Def. 7) ist bereits stabil. Die Frage der Stabilität von partiellen Realisierungen ist in (Byrnes und Lindquist 1982) untersucht worden. Satz 5 liefert ein entsprechendes Resultat für die Stabilisierbarkeit.

Als weitere Anwendung wird jetzt gezeigt, daß sich mit Hilfe der System-Jacobi-Form eine Normalform für die Ähnlichkeitsaktion und statische Ausgangsrückführung (Hinrichsen und Prätzel-Wolters 1984) konstruieren läßt. Definiere dazu

$$J'(n, m, p) := \{(A, B, C) \in J(n, m, p) \mid a_{ij} = 0 \text{ für } (i, j) \in \underline{m} \times \underline{p}\}.$$

Die Systeme in $J'(n, m, p)$ zeichnen sich also dadurch aus, daß die $(m \times p)$ -Nord-West-Ecke von A verschwindet. Es gilt:

Satz 6

Zu jedem $(A, B, C) \in \hat{\Delta}(n, m, p)$ gibt es genau ein $S \in \text{Gl}(n)$ und genau ein $F \in \mathbb{R}^{m \times p}$ mit

$$(S^{-1}(A+BFC)S, S^{-1}B, CS) \in J'(n, m, p). \quad (41)$$

Beweis

Existenz: Wähle S so, daß $(S^{-1}AS, S^{-1}B, CS) \in J(n, m, p)$. Aufgrund der speziellen Struktur von $S^{-1}B$ und CS kann man dann (41) erreichen.

Eindeutigkeit: Falls $S_1, S_2 \in GL(n)$ und $F_1, F_2 \in \mathbb{R}^{m \times p}$ so gewählt sind, daß

$$(A_i, B_i, C_i) := (S_i^{-1}(A + B F_i C) S_i, S_i^{-1}B, C S_i) \in J'(n, m, p),$$

so folgt mit $S := S_1^{-1}S_2$ und $F := F_2 - F_1$ die Identität $(A_2, B_2, C_2) = (S^{-1}(A_1 + B_1 F C_1) S, S^{-1}B_1, C_1 S)$. Da sowohl (A_2, B_2, C_2) , als auch $(A_1 + B_1 F C_1, B_1, C_1)$ in $J(n, m, p)$ enthalten sind, folgt aus Satz 4 ihre Gleichheit und $S = I$. Da die $(m \times p)$ -Nord-West-Ecke sowohl in A_2 als auch in A_1 verschwindet, folgt $F = 0$.

□

Bemerkung 31

Satz 6 ist nicht direkt vergleichbar mit den Resultaten in (Hinrichsen und Prätzel-Wolters 1984). Ein Vorteil von Satz 6 ist, daß er ohne weitere Einschränkungen an n, m und p für generische Systeme eine kanonische Form liefert. Andererseits bricht die Konstruktionsmethode völlig zusammen, falls die Generizitätsannahme verletzt ist. In (Hinrichsen und Prätzel-Wolters 1984) erhält man in diesem Fall zumindest noch eine quasi-kanonische Form.

Mit Hilfe des folgenden Algorithmus läßt sich die System-Jacobi-Form (SJF) berechnen.

Algorithmus SJF

1. Eingabe $n \geq 1, m \geq 1, p \geq 1, (A,B,C) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n}$.
2. Berechne die Kronecker-Hessenberg-Form (A_1, B_1) von (A, B) und transformiere C simultan: $(A_1, B_1, C_1) = (Q^T A Q, Q^T B, C Q)$.
3. Falls (A_1, B_1) nicht die Form

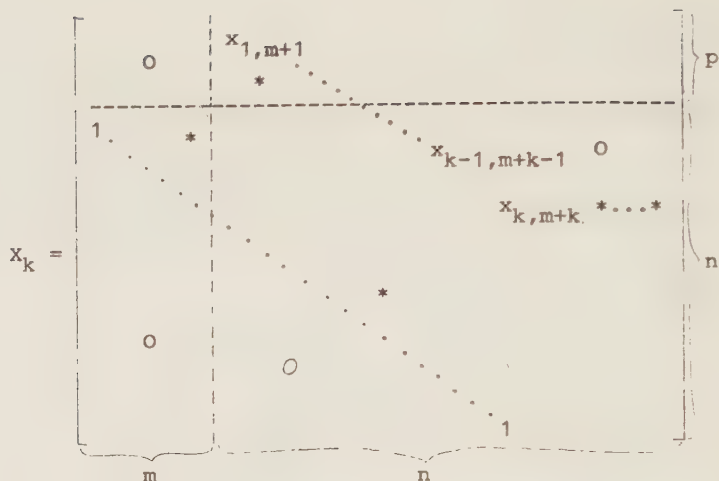
$$B_1 = \begin{bmatrix} \oplus & \dots & * \\ & \ddots & \\ & & \oplus \\ & & & 0 \end{bmatrix}; \quad A_1 = \begin{bmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & & \vdots \\ \oplus & \dots & \vdots \\ & & 0 \\ & & \oplus & \dots & * \end{bmatrix}$$

besitzt, so stoppe $((A, B, C) \notin \mathcal{L}(n, m, p))$.

4. Normiere die Pivotelemente zu 1 durch eine diagonale Ähnlichkeitstransformation: $(A_2, B_2, C_2) = (S^{-1} A_1 S, S^{-1} B_1, C_1 S)$.
5. $k := 1$.
6. Setze

$$X_k := \begin{bmatrix} 0 & C_2 \\ B_2 & A_2 \end{bmatrix}.$$

7. Kommentar: Die Matrix X_k hat die Form



8. Falls $x_{k,m+k} = 0$, so stoppe $((A,B,C) \notin \hat{\Sigma}(n,m,p))$.
9. Falls $k = n$, so stoppe (System-Jacobi-Form ist berechnet).
10. Nutze $x_{k,m+k}$ als Pivotelement und eliminiere die Einträge rechts von $x_{k,m+k}$ durch eine Ähnlichkeitstransformation $S_k \in \text{Ut}^*(n)$.
11. $k \leftarrow k+1$.
12. Gehe zu 7.

Bemerkung 32

Der Algorithmus SJF ist nicht numerisch stabil. Er vermeidet aber zumindest die Aufstellung der Erreichbarkeits-, Beobachtbarkeits- und Hankel-Matrix.

(Schluß-)Bemerkung 33

Mit Hilfe der Lemmata 6 und 7 läßt sich auch eine (weitgehend symmetrische) kanonische Form für generische multi-nomiale Systeme (X,Y) (Hinrichsen 1984) ableiten:

Es seien $X \in \mathbb{R}^{k \times n}$ und $Y \in \mathbb{R}^{n \times \ell}$ ($n \leq k$, $n \leq \ell$) so, daß die ersten n führenden Hauptminoren der Matrix $H := X \cdot Y$ von Null verschieden sind (diese Annahme ist generisch erfüllt). Dann gibt es eine eindeutig bestimmte Matrix $S \in \text{Gl}(n)$, so daß $L := XS$ und $R := S^{-1}Y$ die Form (38) haben.

3.3. Anmerkungen

Verallgemeinerte Zustandsraummodelle der Form

$$E\dot{x} = Fx + Gu \quad (42)$$

mit $E \in \text{Gl}(n)$ und $(FE^{-1}, G) \in \Sigma(n,m)$ sind äquivalent zu einem System (1) mit $(A,B) = (FE^{-1}, G)$. Bei der numerischen Analyse von (42) empfiehlt es sich jedoch nicht, das System (FE^{-1}, G) explizit zu bilden (E^{-1} könnte schlecht konditioniert sein, und außerdem sollte man Produkte von Matrizen grundsätzlich vermeiden). Daher ist die Aktion

$$((S,T), (E,F,G)) \mapsto (SET, SFT, SG) \quad (43)$$

mit $S,T \in \text{O}(n)$, $E \in \text{Gl}(n)$, $(FE^{-1}, G) \in \Sigma(n,m)$ von besonderem numerischen Interesse (Paige 1981; van Dooren 1981; Miminis und Paige 1982). Diese Aktion läßt sich mit den hier

vorgestellten Methoden ebenfalls studieren. So lassen sich z.B. relativ einfach Normalformen bzgl. (43) konstruieren (Laub und Linnemann 1984).

Eine weitere mögliche Erweiterung bezieht sich auf Systeme (A,B) , bei denen die Einträge der Matrizen A und B Elemente eines Hauptidealringes R sind. Die Übersichtsaufsätze (Sontag 1981; Kamen 1978) führen in diese Systemklasse ein und enthalten Motivationen. Diejenigen Zustandsdarstellungsformen, die man bei reellen Systemen durch orthogonale Transformationen erreichen kann, kann man bei Systemen über Hauptidealringe durch (in R) invertierbare Transformationen auch erreichen. Der Schlüssel hierzu ist die Beobachtung, daß zu jedem $(x,y)^T \in \mathbb{R}^2$ eine invertierbare Matrix $Q \in \mathbb{R}^{2 \times 2}$ existiert mit $Q(x,y)^T = (z,0)^T$ und $z = qg^T(x,y)$. Das Konzept einer (nicht notwendig eindeutigen) System-Hessenberg-Form überträgt sich also völlig analog auf Systeme über R . Die $Gl(n)$ -Normalformen kann man allerdings über Ringe i.a. nicht herstellen, da man die Pivotelemente nicht "zu 1 normieren" kann. Stellt man aber zusätzliche Bedingungen an das Ausgangssystem, die sicherstellen, daß die Pivotelemente Einheiten sind, so lassen sich auch diese Resultate übertragen.

4. POLZUWEISUNG DURCH ZUSTANDSRÜCKFÜHRUNG

- 4.1. Zustandsrückführung bei Systemen mit einem Eingang 94
 - 4.1.1. Ein Algorithmus für Frobenius-Systeme 95
 - 4.1.2. Ein Algorithmus für Newton-Systeme 96
 - 4.1.3. Eine Weiterentwicklung 98
 - 4.1.4. En bloc Algorithmen 102
 - 4.1.5. Numerische Erfahrungen 110
- 4.2. Zustandsrückführung bei Systemen mit mehreren Eingängen 113
 - 4.2.1. Sukzessive Algorithmen 114
 - 4.2.2. En bloc Algorithmen und robuste Polzuweisung 119
- 4.3. Großsysteme mit dünn besetzten Systemmatrizen 124
- 4.4. Anmerkungen 128

4.1. Zustandsrückführung bei Systemen mit einem Eingang

In diesem Abschnitt werden Algorithmen für das folgende Problem vorgestellt.

Polzuweisungsproblem

Gegeben seien $(A, b) \in \Sigma(n, 1)$ und $\tilde{s} = (s_1, \dots, s_n) \in \mathbb{C}^n$.
Finde $f \in \mathbb{R}^{1 \times n}$, so daß $\text{spec}(A + bf) = [\tilde{s}]$.

Für dieses Problem existieren bereits eine Reihe von Algorithmen, von denen angenommen wird, daß sie numerisch zuverlässig arbeiten (vgl. Einleitung). Es ist jedoch recht schwierig, das Verhältnis der Algorithmen zueinander zu durchleuchten, da es eine Vielzahl von Möglichkeiten gibt, die Teilschritte eines komplexen Algorithmus zu realisieren. Muß beispielsweise an irgendeiner Stelle eines Polzuweisungsalgorithmus ein lineares Gleichungssystem $Ax = b$ gelöst werden, so kann man die Matrix A auf sehr verschiedene Weisen faktorisieren (z.B. $A = QR$ mit $Q \in O(n)$ und $R = \text{Ut}(n)$ oder $A = \bar{Q} \bar{R}$ mit $\bar{Q} \in O(n)$ und $\bar{R} = \begin{bmatrix} \circ & & \oplus \\ & \ddots & \\ \oplus & & * \end{bmatrix}$, $A = LR$ mit $L = \text{Produkt von stabilisierten elementaren Transformationen}$ $R \in \text{Ut}(n)$ oder $A = U\bar{S}V$ (Singuläre-Werte-Zerlegung) oder...). Ferner kann man die auftretenden orthogonalen Matrizen als Produkt von Givens-Reflektionen, Givens-Rotationen, Householder-Matrizen oder... darstellen, und jedesmal erhält man einen etwas anderen Algorithmus.

In diesem Abschnitt werden systematisch eine Reihe von Algorithmen für das Polzuweisungsproblem entwickelt. Einige dieser Algorithmen sind völlig neu, und andere sind ähnlich bzw. identisch zu bereits bekannten Algorithmen.

Der Leitfaden dabei ist die in der Einleitung beschriebene "Zweischrittmethode". Dabei spielt die Ähnlichkeitstransformation eine besondere Rolle, da $f \in \mathbb{R}^{1 \times n}$ genau dann eine Lösung des Ausgangsproblems ist, wenn $f = \tilde{f}S^{-1}$, wobei $\tilde{f}$ eine Lösung des Polzuweisungsproblems für das System $(S^{-1}AS, S^{-1}b)$ darstellt. Daher können die in Kapitel 4 betrachteten speziellen Zustandsdarstellungsformen zur Vereinfachung des Polzuweisungsproblems herangezogen werden.

Über numerische Erfahrungen mit Polzuweisungsalgorithmen wird im abschließenden Abschnitt 4.1.5. berichtet.

4.1.1. Ein Algorithmus für Frobenius-Systeme

Berechnet man zunächst die System-Frobenius-Form von $(A,b) \in \Sigma(n,1)$, so läßt sich der folgende sehr einfache Algorithmus zur Polzuweisung heranziehen.

Algorithmus PZF

1. Eingabe: $n, (A,b) \in \Sigma(n,1), [\tilde{s}] \in \mathbb{C}_{\text{sym}/\sim}^n$.

2. Berechne die System-Frobenius-Form von (A, b) . Es seien $a_1, \dots, a_n \in \mathbb{C}$ die Koeffizienten und es sei $S \in GL(n)$ die Transformationsmatrix aus Folgerung 2 in Kapitel 3.
3. Berechne die Koeffizienten $\tilde{a}_1, \dots, \tilde{a}_n \in \mathbb{R}$ des gewünschten charakteristischen Polynoms:

$$\prod_{i=1}^n (s - s_i) = \tilde{a}_n + \tilde{a}_{n-1}s + \dots + \tilde{a}_1s^{n-1} + s^n; \tilde{s}^T = (s_1, \dots, s_n).$$
4. Berechne die Rückführung

$$f = (-\tilde{a}_1 + a_1, \dots, -\tilde{a}_n + a_n)S^{-1}.$$
5. Stoppe.

Dieser Algorithmus ist schon sehr lange bekannt (Kalman 1963) und galt lange Zeit als beste Methode zur Lösung des Polzuweisungsproblems.

Man beachte, daß es in Schritt 4 nicht notwendig ist, die Matrix S zu invertieren. Vielmehr kann S^{-1} während der Berechnung der System-Frobenius-Form sukzessive aufgebaut werden.

4.1.2. Ein Algorithmus für Newton-Systeme

Ausgehend von der System-Newton-Form mit Stützstellen $s_1, \dots, s_n$ erhält man den folgenden Algorithmus.

Algorithmus PZN

1. Eingabe $n, (A, b) \in \Sigma(n, 1), [\tilde{s}] \in \mathbb{C}_{\text{sym}/\sim}^n$.
2. Berechne die (komplexe) System-Newton-Form mit Stützstellen $s_1, \dots, s_n$ von (A, b) , wobei $\tilde{s} = (s_1, \dots, s_n)^T$. Es seien $a_1, \dots, a_n \in \mathbb{C}$ die Koeffizienten und es sei $S \in \text{Gl}(n, \mathbb{C})$ die Transformationsmatrix aus Proposition 2 in Kapitel 3.
3. Berechne die Rückführung
$$f = (a_1, \dots, a_n) S^{-1}.$$
5. Stoppe.

Wie bei Algorithmus PZF muß auch hier die Matrix S nicht explizit invertiert werden.

Der Algorithmus PZN ist in komplexer Arithmetik durchzuführen. Das Endergebnis f ist jedoch wieder reell. Die komplexe Arithmetik läßt sich durch Heranziehung der reellen Newton-Form vermeiden. Der dabei entstehende Algorithmus entspricht dem Algorithmus in (Eising 1982), spezialisiert auf Systeme mit einem Eingang.

Man beachte, daß durch den Algorithmus PZN ein Jordan-System entsteht, und daß für $s_1 = \dots = s_n = 0$ der Algorithmus PZF entsteht.

4.1.3. Eine Weiterentwicklung

Die System-Newton-Form läßt sich nicht numerisch stabil berechnen. Daher wird hier eine Weiterentwicklung des Algorithmus PZN vorgestellt, bei der nur unitäre Ähnlichkeits- und Rückführungsoperationen benutzt werden. Wie in Abschnitt 4.1.2. wird wieder die komplexe Fassung dargestellt; es entsteht dann der folgende Algorithmus PZS (Polzuweisung durch sukzessives Verschieben der Eigenwerte).

Algorithmus PZS

1. Eingabe: $n \geq 2$, $(A, b) \in H(n, 1)$, $[\tilde{s}] \in \mathbb{C}_{\text{sym}}^n / \sim$.
2. Setze $k := n$; $A^k := A$; $b^k := b$; $Q^k := I \in U(n, \mathbb{C})$; $f^k \in \mathbb{R}^{1 \times n}$.
3. Kommentar (vgl. anschließenden Beweis): Es gilt

$$(i) \quad k \in \{0, 1, 2, \dots, n\}$$

$$(ii) \quad A^k = (Q^k)^H (A + b f^k) Q^k; \quad b^k = (Q^k)^H b; \quad Q^k \in U(n, \mathbb{C})$$

$$(iii) \quad A^k = \left[\begin{array}{c|c} \begin{array}{ccc} * & \dots & * \\ \oplus & & \vdots \\ & \ddots & * \\ O & & \oplus \\ & & * \end{array} & \begin{array}{c} \vdots \\ O \\ \vdots \end{array} \\ \hline * & \begin{array}{c} s_{k+1} \\ \vdots \\ s_n \end{array} \end{array} \right] \begin{array}{l} \left. \vphantom{\begin{array}{c} \vdots \\ O \\ \vdots \end{array}} \right\} k \\ \left. \vphantom{\begin{array}{c} * \\ \vdots \\ s_n \end{array}} \right\} k \end{array} ; \quad b^k = \left[\begin{array}{c} * \\ \vdots \\ O \\ * \end{array} \right] \begin{array}{l} \left. \vphantom{\begin{array}{c} * \\ \vdots \\ O \end{array}} \right\} k \\ \left. \vphantom{*} \right\} \end{array} .$$

4. Falls $k = 0$, so stoppe ($f := f^0$ ist die gewünschte Rückführung).

5. Bilde $A^k - s_k I$ und wende $k - 1$ Givens-Rotationen auf diese Matrix an, die jeweils die Spalten $(k-1 \text{ und } k), (k-2 \text{ und } k), \dots, (1 \text{ und } k)$ dieser Matrix beeinflussen und die die Einträge in den Positionen $(k,k), (k-1,k), \dots, (2,k)$ auslöschen. Die Givens-Rotationen fasse zu einer unitären $(n \times n)$ -Matrix $\bar{Q}^k$ zusammen.

6. Kommentar: Es gilt

$$\bar{A}^k := (A^k - s_k I) \bar{Q}^k = \left[\begin{array}{ccc|ccc} * & \dots & * & & & \\ \oplus & & & & & \\ & \ddots & & & & \\ & & * & & & \\ 0 & & \oplus & 0 & & \\ & & & & & \\ \hline & & & s_{k+1} - s_k & & \\ & * & & & \ddots & \\ & & & & & s_n - s_k \end{array} \right]$$

7. Setze $\bar{f}^k := [0 \dots 0 \overset{k}{\xi^k} 0 \dots 0]$, wobei $\xi^k := -d^k/\beta^k$, und d^k den $(1,k)$ -ten Eintrag von $\bar{A}^k$ und β^k den ersten Eintrag in b^k bezeichnet.

8. Berechne

$$A^{k-1} := (\bar{Q}^k)^H (\bar{A}^k + b^k \bar{f}^k) + s_k I$$

$$b^{k-1} := (\bar{Q}^k)^H b^k$$

$$Q^{k-1} := Q^k \bar{Q}^k$$

$$f^{k-1} := f^k + \bar{f}^k (\bar{Q}^k)^H (Q^k)^H.$$

9. $k \leftarrow k-1$.

10. Gehe nach 3.

Der Algorithmus stoppt also mit einem System das aus dem ursprünglichen System durch unitäre Ähnlichkeits- und Rückführungsoperationen hervorgeht. Ferner ist die A-Matrix in Schur - Form — mit den gewünschten Eigenwerten auf der Diagonalen.

Bemerkung 1

Der Algorithmus PZS unterscheidet sich vom Algorithmus PZN im wesentlichen nur dadurch, daß in Schritt 5 anstatt der (nicht-unitären) Transformationen der Form

$$S^k = \begin{bmatrix} 1 & 0 & * & & \\ & \ddots & \vdots & & \\ & & * & & \\ & & & 1 & 0 \\ & 0 & & & \ddots & 1 \end{bmatrix}$$

unitäre Givens-Transformationen eingesetzt werden. Dadurch wird ein Teil der Struktur des Systems zerstört (es entstehen die i.a. von Null verschiedenen Parameter im unteren Teil von b^k und in der SW-Ecke von A^k), die "wesentliche" Struktur bleibt jedoch erhalten.

Bemerkung 2

Bei einer reellen Version des Algorithmus PZS müßte jeweils ein Paar konjugiert komplexer Pole gleichzeitig behandelt werden, ähnlich wie dieses bei der reellen Version des QR-Algorithmus (Stewart 1973) geschieht.

Bemerkung 3

Für $k = n$ wird im Algorithmus PZS eine QR-Zerlegung der Form

$$A - s_n I = \begin{bmatrix} * & \cdot & \cdot & \cdot & * & * \\ \oplus & & & & & 0 \\ & \cdot & \cdot & \cdot & & \cdot \\ & & \cdot & * & & \cdot \\ & & & & \oplus & 0 \end{bmatrix} \cdot (\bar{Q}^n)^H$$

der Matrix $A - s_n I$ durchgeführt. Eine QR-Zerlegung der Form

$$(A - s_n I) = \begin{bmatrix} * & & & * \\ & \oplus & & \\ 0 & & \cdot & \\ & & & \oplus \end{bmatrix} \cdot (\bar{Q}^n)^H$$

erhält man, wenn man die Givens-Transformationen auf die Spalten $(n-1 \text{ und } n)$, $(n-2 \text{ und } n-1), \dots, (1 \text{ und } 2)$ anwendet. Eine entsprechende Änderung des Algorithmus ist prinzipiell möglich, und in Schritt 3 würde man Matrizen der Form

$$\begin{bmatrix} s_1 & & & * & & * \\ & \cdot & & & & \\ 0 & & \cdot & & & \\ & & & s_k & & \\ \hline & & & & \cdot & \cdot & \cdot & * \\ 0 & & & & \oplus & \cdot & \cdot & \cdot \\ & & & & & \cdot & \oplus & * \end{bmatrix}$$

erhalten, wobei der NW-Block bereits gewünschte Eigenwerte enthält. In dieser Version ist der Algorithmus sehr ähnlich zu dem von *Mimimis* und *Paige* (1982), er ist jedoch viel übersichtlicher, da nicht mit "unbekannten Zeilen" operiert wird.

4.1.4. En bloc Algorithmen

Die Polzuweisung in den Algorithmen PZF und PZN geschieht en bloc, d.h. alle Pole werden gleichzeitig in die gewünschten Positionen verschoben. Im Algorithmus PZS werden die Pole dagegen sukzessiv verschoben. Der Algorithmus PZN lässt sich auch derart modifizieren, daß die Pole sukzessiv verschoben werden. Der Algorithmus PZF ist jedoch von seiner Konzeption her ein "en bloc Algorithmus", genauso wie PZS ein "sukzessiver Algorithmus" ist. In diesem Abschnitt werden en bloc Algorithmen vorgestellt, bei denen die numerisch instabilen Operationen zur Herstellung der Frobenius-Form vermieden werden. Die Algorithmen beruhen auf dem folgenden Lemma.

Lemma 1

Es sei $(A, b) \in \Sigma(n, 1)$ und es seien $s_1, \dots, s_n \in \mathbb{C}^n$ paarweise verschieden. Für alle $i \in \underline{n}$ sei $0 \neq [d_i, r_i^T]^T$ mit $r_i \in \mathbb{C}^n$ und $d_i \in \mathbb{C}$ eine Lösung der Gleichung

$$\begin{bmatrix} b & A - s_i I \end{bmatrix} \begin{bmatrix} d_i \\ r_i \end{bmatrix} = 0. \quad (1)$$

Dann ist $f \in \mathbb{C}^{1 \times n}$ genau dann eine Lösung des Polzuweisungsproblems, wenn

$$f r_i = d_i \quad \text{für alle } i \in \underline{n}. \quad (2)$$

Beweis

Falls f eine Lösung des Polzuweisungsproblems ist, so gibt es $0 \neq z_i \in \mathbb{C}^n$ mit $(A - s_i I + b f) z_i = 0$ für $i \in \underline{n}$, d.h. es gilt

$$[b \mid A-s_i I] \begin{bmatrix} fz_i \\ z_i \end{bmatrix} = 0 \quad \text{für } i \in \underline{n}.$$

Aufgrund der Erreichbarkeit von (A,b) ist die Lösung von (1) bis auf skalare Vielfache eindeutig bestimmt, und daher folgt $d_i = \alpha_i fz_i$ und $r_i = \alpha_i z_i$ für geeignete $0 \neq \alpha_i \in \mathbb{C}$. Dieses impliziert (2). Sind andererseits (1) und (2) erfüllt, so folgt $r_i \neq 0$ (da $b \neq 0$) und

$$(A-s_i I + bf)r_i = [b \mid A-s_i I] \begin{bmatrix} d_i \\ r_i \end{bmatrix} = 0,$$

d.h. f ist eine Lösung des Polzuweisungsproblems. $\square$

Bemerkung 4

Falls (A,b) erreichbar ist und (1) gilt, so ist (2) eindeutig lösbar und r_i ist ein Eigenvektor des geschlossenen Systems. Ferner, falls $s = (s_1, \dots, s_n) \in \mathbb{C}_{\text{sym}}^n$, so ist die Lösung f von (2) reell.

Bemerkung 5

Falls $s_j = s_i^H$ und $[d_i, r_i^T]^T$ die Gleichung (1) löst, so ist $[d_j, r_j^T]^T := [d_i^H, r_i^H]^T$ ebenfalls Lösung (für s_j). Ferner sind die Gleichungen

$$fr_i = d_i; \quad f(r_i^H)^T = d_i^H; \quad f \in \mathbb{R}^{1 \times n}$$

äquivalent zu

$$f(\operatorname{Re}(r_i)) = \operatorname{Re}(d_i); \quad f(\operatorname{Im}(r_i)) = \operatorname{Im}(d_i).$$

Aus Lemma 1 und den Bemerkungen 4 und 5 ergibt sich unmittelbar die Rechtfertigung für den folgenden Algorithmus PZEB (Polzuweisung durch en bloc-Verschiebung der Eigenwerte).

Algorithmus PZEB

1. Eingabe: $n, (A, b) \in \Sigma(n, 1), s = (s_1, \dots, s_n) \in \mathbb{C}_{\text{sym}}^n$
mit $s_i \neq s_j$ für $i \neq j$.
2. Für alle $i \in \underline{a} := \underline{a}(s)$ löse die Gleichungssysteme

$$\begin{bmatrix} b & A - s_{2i-1}I \end{bmatrix} \begin{bmatrix} d_i \\ r_i \end{bmatrix} = 0; \quad \begin{bmatrix} d_i \\ r_i \end{bmatrix} \neq 0$$

in komplexer Arithmetik und setze

$$\begin{aligned} x_{2i-1} &:= (\operatorname{Re}(r_i))^T \in \mathbb{R}^{1 \times n} \\ x_{2i} &:= (\operatorname{Im}(r_i))^T \in \mathbb{R}^{1 \times n} \\ y_{2i-1} &:= \operatorname{Re}(d_i) \in \mathbb{R} \\ y_{2i} &:= \operatorname{Im}(d_i) \in \mathbb{R}. \end{aligned}$$

3. Für alle $i = 2a + 1, \dots, n$ löse die Gleichungssysteme

$$\begin{bmatrix} b_i & A - s_i I \end{bmatrix} \begin{bmatrix} d_i \\ r_i \end{bmatrix} = 0; \quad \begin{bmatrix} d_i \\ r_i \end{bmatrix} \neq 0$$

in reeller Arithmetik und setze

$$\begin{aligned} x_i &:= r_i^T \in \mathbb{R}^{1 \times n} \\ y_i &:= d_i \in \mathbb{R}. \end{aligned}$$

4. Bilde

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}; \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

und löse das Gleichungssystem

$$Xf^T = y.$$

5. Stoppe.

Bemerkung 6

Die Gleichungssysteme in den Schritten 2 bzw. 3 lassen sich beispielsweise mit Hilfe der QR-Zerlegung lösen. Gilt nämlich

$$[b \mid A - s_i I] = \begin{bmatrix} 0 & \oplus & \dots & * \\ \vdots & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & \oplus \end{bmatrix} \cdot Q^H, \quad (3)$$

so ist $[d_i, r_i^T]^T$ gegeben durch die erste Spalte von Q . Alternativ kann die Singuläre-Werte-Zerlegung oder die LU-Zerlegung herangezogen werden. Da das Lösen des Gleichungssystems in Schritt 4 auch ein gut studiertes Problem der numerischen linearen Algebra ist, können sämtliche Schritte des Algorithmus PZEB mit Hilfe von Standardprogrammen (etwa die allgemein als sehr zuverlässig anerkannten Programme in dem Programmpaket LINPACK; vgl. Dongarra u.a. 1979) ausgeführt werden.

Bemerkung 7

Der Algorithmus PZEB arbeitet mit keiner speziellen Zustands-

darstellungsform; er ist also nicht der Zweischrittmethode zuzuordnen. Dieses hat den Vorteil, daß der Algorithmus auch noch zuverlässig arbeiten kann, wenn die System-Hessenberg-Form schlecht konditioniert ist. Allerdings sind (im Falle von reellen Eigenwerten s_i) $n+1$ voll besetzte Gleichungssysteme zu lösen. Die folgende Bemerkung zeigt, daß sich dieser rechnerische Aufwand erheblich reduzieren läßt, wenn man die System-Hessenberg-Form benutzt, also doch die Zweischrittmethode anwendet.

Bemerkung 8

Falls $(A,b) \in H(n,1)$, so haben die Gleichungssysteme in den Schritten 2 und 3 die Form

$$\begin{bmatrix} \oplus & * & \dots & * \\ & \cdot & & \vdots \\ & & \ddots & \vdots \\ & & & \oplus & * \end{bmatrix} \begin{bmatrix} d_i \\ r_i \end{bmatrix} = 0. \quad (4)$$

Eine QR-Zerlegung der entsprechenden Matrix erfordert lediglich n Givens-Transformationen, und das Gleichungssystem (4) läßt sich sehr effektiv und zuverlässig (Wilkinson 1961) lösen. Alternativ zur QR-Zerlegung kann man die Matrix in (4) auch wie folgt aufteilen

$$\left[\begin{array}{c|c} b & A-s_i I \end{array} \right] = \left[\underbrace{\bar{A}_i}_n \mid \underbrace{\bar{b}_i}_1 \right]$$

und das trianguläre Gleichungssystem $\bar{A}_i \bar{x}_i = -\bar{b}_i$ lösen. Es gilt dann (bis auf skalare Vielfache)

$$\begin{bmatrix} d_i \\ r_i \end{bmatrix} = \begin{bmatrix} \bar{x}_i \\ 1 \end{bmatrix}. \quad (5)$$

Die Rundungsfehleranalyse aus (Wilkinson 1965; Abschn. 7.11-7.15) läßt sich direkt auf den vorliegenden Fall übertragen und zeigt, daß die Zurückführung auf ein trianguläres Gleichungssystem der QR-Zerlegung vorzuziehen ist.

Bemerkung 9

Ist $(A,b) \in H(n,1)$ und teilt man die Matrix in (4) wie folgt auf

$$[b \mid A-s_i I] = \begin{bmatrix} * & * & * \\ 0 & \underbrace{T_i}_{1 \times n-1} & \underbrace{h_i}_1 \end{bmatrix} \begin{matrix} 1 \\ n-1 \\ 1 \end{matrix},$$

so gilt für die Lösung $\bar{r}_i$ des triangulären Gleichungssystems

$$T_i \bar{r}_i = -h_i \quad (6)$$

und für die Lösung von (4) bis auf skalare Vielfache die Identität

$$r_i = \begin{bmatrix} \bar{r}_i \\ 1 \end{bmatrix}, \quad \text{d.h. der Eigenvektor } r_i \text{ des geschlossenen Kreises}$$

ist bestimmt, nicht aber d_i . Das Lösen der Gleichungssysteme

(6) ist der erste Schritt des Polzuweisungsalgorithmus in

(Petkov, Konstantinov und Christov 1982). Da dabei aber der

einfache Zusammenhang aus Lemma 1 übersehen wurde, sind die

weiteren Schritte in (Petkov u.a. 1982) sehr unübersichtlich

und aufwendig.

Bemerkung 10

Ist $(A,b) \in H(n,1)$ und ist $\begin{bmatrix} d_i \\ r_i \end{bmatrix} \neq 0$ eine Lösung von (1), so folgt

$$(A - s_i I) R_i = \begin{bmatrix} * & \dots & * & * \\ \oplus & \dots & \dots & 0 \\ & \dots & \dots & \vdots \\ 0 & & \dots & \oplus & 0 \end{bmatrix}, \quad (7)$$

wobei

$$R_i := \left[\begin{array}{c|c} -I & x_i \\ \hline 0 & 1 \end{array} \right] \begin{matrix} \} n-1 \\ \} 1 \end{matrix} \in \text{Ut}(n).$$

$\underbrace{\hspace{1.5cm}}_{n-1} \quad \underbrace{\hspace{1.5cm}}_1$

Dieses erhellt den Zusammenhang zwischen den Algorithmen PZS und PZEB: In PZS wird zunächst die Form (7) für $i = n$ gebildet, dann s_n verschoben und schließlich das Problem um eine Dimension reduziert (sukzessives Verschieben der Eigenwerte). In PZEB wird die Form hingegen zunächst für alle $i \in \underline{n}$ gebildet, und schließlich werden alle Eigenwerte en bloc verschoben.

Bemerkung 11

Ein Algorithmus, der nicht auf Hessenberg-Formen aufbaut und bei dem die Eigenwerte ebenfalls en bloc verschoben werden, findet sich in (Konstantinov, Petkov und Christov 1981a).

Bemerkung 12

Im nächsten Abschnitt wird sich zeigen, daß auch der Algorithmus PZS so modifiziert werden kann, daß er die System-Hessenberg-Form vermeidet.

Bemerkung 13

Aus mehreren Teilschritten bestehende Algorithmen sind häufig

nicht numerisch stabil. Dieses liegt daran, daß die Hintereinanderschaltung und Parallelschaltung von numerisch stabilen Teilalgorithmen i.a. nicht wieder einen numerisch stabilen Algorithmus ergeben. So läßt sich zwar jeder Teilschritt des Algorithmus PZEB numerisch stabil implementieren, was aber noch nicht impliziert, daß der Gesamtalgorithmus numerisch stabil ist. Es läßt sich jedoch eine Vorwärtsanalyse anstellen, die besagt, daß der Algorithmus - unter gewissen Voraussetzungen an die Daten - zuverlässig arbeitet. Diese Analyse wird hier nicht exakt durchgeführt; vielmehr wird der wesentliche Beweisgang umgangssprachlich erläutert. Zunächst zu den Voraussetzungen: Es wird angenommen, daß das System "gut" erreichbar ist, in dem Sinne, daß dessen Entfernung zu einem nicht erreichbaren System nicht zu klein ist, d.h. (Eising 1984) daß der kleinste singuläre Wert von $[b \mid A-sI]$ genügend groß ist für alle $s \in \mathbb{C}$. Ferner wird angenommen, daß $s_1, \dots, s_n$ so gewählt sind, daß sie - als Eigenwerte des geschlossenen Systems - nicht zu sensitiv auf Störungen reagieren, d.h. (Wilkinson 1965; Abschn. 2.30), daß $\kappa(\tilde{X})$ genügend groß ist, wobei die Spalten von $\tilde{X}$ die Eigenvektoren von $A+bf$ bezüglich der s_i sind und $\kappa(\tilde{X}) := |\tilde{X}|_2 \cdot |\tilde{X}^{-1}|_2$. Jetzt läßt sich die folgende Argumentationskette führen: Da die Gleichungssysteme in den Schritten 2 und 3 des Algorithmus PZEB nach Voraussetzung gut konditioniert sind, läßt sich die Lösung $\begin{bmatrix} d_i \\ r_i \end{bmatrix}$ von (1) mit kleinen Rundungsfehlern berechnen. Damit treten in X und y in Schritt 4 auch nur kleine Fehler auf. Da $\kappa(\cdot)$ gut konditioniert ist (Stewart 1973, Theorem 6.6) und $\kappa(\tilde{X})$ nach Voraussetzung nicht zu klein ist,

ist auch das berechnete Gleichungssystem in Schritt 4 des Algorithmus PZEB gut konditioniert. Damit läßt sich f zuverlässig berechnen, und die tatsächlich zugewiesenen Eigenwerte sind nahe bei den $s_{i \cdot (c)}$. Eine exakte Formulierung dieser Bemerkung steht noch aus.

4.1.5. Numerische Erfahrungen

Die folgenden vier Algorithmen zur Polzuweisung durch Zustandsrückführung bei Systemen mit einem Eingang wurden implementiert und vergleichend auf Testprobleme angewendet.

- Algorithmus PZF
- Algorithmus PZN (komplexe Arithmetik)
- Algorithmus PZEB (der Einfachheit halber wurde angenommen, daß $s_1, \dots, s_n$ reell sind, so daß Schritt 2 entfällt. Es wurde eine Version implementiert, bei der zunächst die System-Hessenberg-Form und dann $\begin{bmatrix} d_i \\ r_i \end{bmatrix}$ mittels (5) berechnet wurde (vgl. Bemerkung 8). Die Teilschritte des Algorithmus wurden realisiert durch die Unterprogramme ORTHES, ORTBK, STRCO, STRSL, SGEKO, SGEFA und SGESL aus EISPACK und LINPACK (Smith u.a. 1976; Dongarra u.a. 1979))
- Algorithmus PZMP¹⁾ (der Algorithmus von Miminis und Paige (1982) Paare konjugiert komplexer Eigenwerte werden durch einen "Doppelschritt" in reeller Arithmetik allokiert).

¹⁾ Dieser Algorithmus ist von G. S. Miminis und C. C. Paige zur Verfügung gestellt worden. Für die freundliche Unterstützung möchte ich mich recht herzlich bedanken.

Experiment 1

Ausgegangen wurde von Daten der Form $A = \text{diag}(s_1, \dots, s_n)$; $b = (b_1, \dots, b_n)^T$; $f = (f_1, \dots, f_n)$, wobei $s_i, b_i, f_i \in [-10, 10]$ durch den Pseudo-Zufallszahlengenerator in (Smith u.a. 1976) berechnet wurden. Sodann wurde $\tilde{A} := A - bf$ gesetzt, und die Polzuweisungsalgorithmen wurden beauftragt, die Eigenwerte von $(\tilde{A}, b)$ nach $s_1, \dots, s_n$ zu verschieben. Die exakte Lösung ist $f = (f_1, \dots, f_n)$, und die folgende Tabelle gibt die Abweichung (Maximum-Norm) zwischen f und der berechneten Rückführung für verschiedene n an.

	PZF	PZN	PZEB	PZMP
$n = 10$	$0.1 \cdot 10^{-1}$	$0.8 \cdot 10^{-3}$	$0.9 \cdot 10^{-4}$	$0.3 \cdot 10^{-3}$
$n = 20$	$0.6 \cdot 10^2$	$0.2 \cdot 10^{-1}$	$0.7 \cdot 10^{-3}$	$0.1 \cdot 10^{-3}$
$n = 30$	$0.5 \cdot 10^6$	$0.5 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$	$0.3 \cdot 10^{-3}$

Tabelle 1

Man erkennt, daß die Algorithmen in der Reihenfolge PZF - PZN - PZEB - PZMP jeweils besser werden. Der Algorithmus PZF ist nur für kleine n ($n < 10$), und die Algorithmen PZN, PZEB sind auch für mittlere n ($n < 30$) anwendbar. Der Algorithmus PZMP liefert sogar für $n = 30$ noch gute Ergebnisse. Die erstaunlich schlechten Ergebnisse bei dem Algorithmus PZF kommen zustande, da in den Zwischenergebnissen (in der System-Frobenius-Form; vgl. die Experimente zu Kapitel 3) sehr große Zahlen auftreten. Durch Tabelle 1 wird die Entwicklung von den (zum Algorithmus PZF) alternativen Algorithmen nachträglich motiviert.

Experiment 2

Ausgegangen wurde von einem System $(A,b) \in [-1,1]^{n \times n} \times [-1, 1]^n$ und von zu setzenden Eigenwerten $s_1, \dots, s_n \in [-2,0]$.

Die Parameter wurden wieder durch den Pseudo-Zufallszahlengenerator aus (Smith u.a. 1976) bestimmt. Sodann wurden die Polzuweisungsalgorithmen beauftragt, die Eigenwerte von A nach $s_1, \dots, s_n$ zu verschieben, und schließlich wurde zur Kontrolle das Spektrum des geschlossenen Systems $(A+bf, b)$ berechnet (mit dem Algorithmus HQR aus EISPACK). Die folgende Tabelle zeigt für verschiedene n die Differenz (Maximum-Norm) zwischen den berechneten Eigenwerten von $A+bf$ und $s_1, \dots, s_n$. In der letzten Spalte ist zusätzlich die Größenordnung (Maximum-Norm) der berechneten Rückführungen angegeben.

	PZF	PZN	PZEB	PZMP	$ f $
$n = 4$	$0.2 \cdot 10^{-1}$	$0.2 \cdot 10^{-1}$	$0.2 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$	$0.4 \cdot 10^1$
$n = 6$	$0.3 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$	$0.1 \cdot 10^0$	$0.1 \cdot 10^0$	$0.2 \cdot 10^2$
$n = 8$	$0.7 \cdot 10^0$	$0.8 \cdot 10^0$	$0.9 \cdot 10^0$	$0.3 \cdot 10^0$	$0.5 \cdot 10^2$
$n = 30$	$0.6 \cdot 10^2$	$0.4 \cdot 10^3$	$0.8 \cdot 10^1$	$0.1 \cdot 10^2$	$0.6 \cdot 10^5$

Tabelle 2

Man beobachtet also sehr schlechte Ergebnisse bei allen Algorithmen. Dieses liegt daran, daß die relativ großen Zahlen in der Rückführung auch große absolute Fehler verursachen. Man beachte, daß man Fehler der Größenordnung aus Tabelle 1 erhält, wenn man die Fehler aus Tabelle 2 durch $|f|$ teilt (relativer Fehler!). Man ist aber an den absoluten Fehlern

interessiert, und daher kann man sagen, daß das *Polzuweisungsproblem für Systeme mit einem Eingang "in der Regel" schlecht konditioniert ist* in dem Sinne, daß große Rückführverstärkungen auftreten, die eine exakte Verschiebung der Eigenwerte unmöglich machen.

Glücklicherweise haben Großsysteme häufig mehrere Eingänge, und die entstehenden Freiheiten können (und müssen) genutzt werden, um die Rückführverstärkungen zu minimieren. Kleine Rückführverstärkungen sind aus regelungstheoretischer Sicht wünschenswert, und die Experimente zeigen, daß sie auch aus numerischer Sicht eine Notwendigkeit sind.

4.2. Zustandsrückführung bei Systemen mit mehreren Eingängen

Prinzipiell lassen sich, mittels Heymann's Lemma (Heymann 1968; Hautus 1977), die für skalare Systeme entworfenen Algorithmen auch zur Polzuweisung bei multivariablen Systemen heranziehen. Dieses läßt sich algorithmisch besonders elegant mit Hilfe der Hermite-Hessenberg-Form (vgl. Kapitel 3) realisieren (Nour Eldin 1974). Aufgrund der Erfahrungen aus Abschnitt 4.1.5. ist dieses jedoch nicht unbedingt zu empfehlen.

In diesem Abschnitt wird gezeigt, daß sich die beiden wesentlichen Typen von Algorithmen (en bloc und sukzessiv) auf den multivariablen Fall übertragen lassen. Die sukzessiven Algorithmen haben dabei den Vorteil, daß sie problemlos verallgemeinerbar sind, und daß man auch im multivariablen Fall einen "algebraischen" Algorithmus erhält (Abschn. 4.2.1.). Bei den en bloc Algorithmen

ist die Verallgemeinerung nicht ohne weiteres möglich, jedoch können durch den Einsatz von Optimierungsmethoden zusätzliche Anforderungen an den Entwurf, wie etwa Robustheit der Eigenwerte des geschlossenen Kreises und Kontrollgrößenminimierung, mit einbezogen werden (Abschn. 4.2.2.).

In Abschnitt 4.3. wird eine etwas andere *Problemstellung* behandelt. Während ansonsten immer vorausgesetzt wird, daß das System keine ausnutzbare Struktur besitzt (vgl. Einleitung), wird in Abschnitt 4.3. das Polzuweisungsproblem für Großsysteme mit dünn besetzten Systemmatrizen behandelt.

4.2.1. Sukzessive Algorithmen

In Bemerkung 12 wurde bereits angedeutet, daß der Algorithmus PZS so modifiziert werden kann, daß er nicht auf eine spezielle Zustandsdarstellungsform aufbaut. Diese Modifikation ist in dem folgenden Algorithmus PZSM (Polzuweisung durch sukzessives Verschieben der Eigenwerte in multivariablen Systemen) vorgenommen.

Algorithmus PZSM

1. Eingabe $n, m \geq 1$, $(A, B) \in \Sigma(n, m)$, $s_1, \dots, s_n \in \mathbb{C}$.
2. $k = n$, $S^k = I$, $F^k = O$, $A^k = A$, $B^k = B$.
3. Kommentar: Es gilt (siehe anschließender Beweis):

(i) $k \in \{0, 1, \dots, n\}$

(ii) $A^k = (S^k)^{-1} (A + BF^k) S^k$

$B^k = (S^k)^{-1} B$

$S^k \in GL(n, \mathbb{C}), F \in \mathbb{C}^{m \times n}$

(iii) A^k und B^k haben die Struktur

$$A^k = \left[\begin{array}{c|c} \bar{A}^k & 0 \\ \hline * & \tilde{A}^k \end{array} \right] \}_{k, \quad B^k = \left[\begin{array}{c} \bar{B}^k \\ \hline * \end{array} \right] \}_{k},$$

$\underbrace{\hspace{1.5cm}}_k$

wobei $(\bar{A}^k, \bar{B}^k) \in \Sigma(k, m)$ (falls $k > 0$) und

$$\tilde{A}^k = \left[\begin{array}{ccc} s_{k+1} & & 0 \\ & \ddots & \\ & * & s_n \end{array} \right] \quad (\text{falls } k < n).$$

Falls $k = 0$, so stoppe, (es gilt $\text{spec}(A + BF^0) = [(s_1, \dots, s_n)]$).

Bilde

$$[\bar{B}^k \mid \bar{A}^k - s_k I]$$

und bestimme, $d^k \in \mathbb{C}^m, r^k \in \mathbb{C}^k$, mit

$$[\bar{B}^k \mid \bar{A}^k - s_k I] \begin{bmatrix} d^k \\ r^k \end{bmatrix} = 0, \quad r^k \neq 0.$$

Kommentar: Da man o.E. annehmen kann, daß $\bar{B}^k$ vollen Spaltenrang besitzt, kann man in Schritt 5 immer $r^k \neq 0$ erreichen.

Bilde eine Matrix $\bar{R}^k \in GL(k, \mathbb{C})$ so, daß r^k die letzte Spalte von $\bar{R}^k$ ist.

8. Berechne

$$R^k = \begin{bmatrix} \bar{R}^k & 0 \\ 0 & I \end{bmatrix} \in Gl(n, \mathbb{C})$$

$$S^{k-1} = S^k R^k$$

$$\bar{F}^k = \underbrace{[0 \dots 0 \quad d^k \quad 0 \dots 0]}_k (R^k)^{-1}$$

$$F^{k-1} = F^k + \bar{F}^k (S^k)^{-1}$$

$$A^{k-1} = (R^k)^{-1} (A^k + B^k \bar{F}^k) R^k$$

$$B^{k-1} = (R^k)^{-1} B^k.$$

9. $k \leftarrow k-1.$

10. Gehe zu 3.

Beweis der Behauptungen in Schritt 3

Für $k = n$ sind die Bedingungen erfüllt. Die Bedingungen seien jetzt erfüllt für ein festes k mit $1 \leq k \leq n$.

Dann gilt

$$\begin{aligned} A^{k-1} &= (R^k)^{-1} (A^k + B^k \bar{F}^k) R^k \\ &= (R^k)^{-1} ((S^k)^{-1} (A + B F^k) S^k + (S^k)^{-1} B \bar{F}^k) R^k \\ &= (S^{k-1})^{-1} (A + B (F^k + \bar{F}^k (S^k)^{-1})) S^{k-1} \\ &= (S^{k-1})^{-1} (A + B F^{k-1}) S^{k-1}, \end{aligned}$$

$$\begin{aligned} B^{k-1} &= (R^k)^{-1} B^k \\ &= (R^k)^{-1} (S^k)^{-1} B \\ &= (S^{k-1})^{-1} B, \end{aligned}$$

. es ist nur noch (iii) zu zeigen. Es gilt jedoch

$$\begin{aligned}
 A^{k-1} &= (R^k)^{-1} (A^k + B^k \bar{F}^k) R^k \\
 &= (R^k)^{-1} \{ (A^k - s_k I) R^k + B^k \bar{F}^k R^k \} + s_k I \\
 &= (R^k)^{-1} \left\{ \left[\begin{array}{c|c} (\bar{A}^k - s_k I) \bar{R}^k & 0 \\ \hline * & \tilde{A}^k - s_k I \end{array} \right] + \left[\begin{array}{c|c} \bar{B}^k & \\ \hline * & \end{array} \right] \cdot [0 \dots 0 \ d^k \ 0 \dots 0] \right\} + s_k I \\
 &= (R^k)^{-1} \left\{ \left[\begin{array}{c|c} * | (\bar{A}^k - s_k I) R^k & 0 \\ \hline * & \tilde{A}^k - s_k I \end{array} \right] + \left[\begin{array}{c|c} 0 & \bar{B}^k d^k \\ \hline 0 & * \end{array} \right] \right\} + s_k I \\
 &= (R^k)^{-1} \left\{ \left[\begin{array}{c|c} * & 0 \\ \hline * & \tilde{A}^k - s_k I \end{array} \right] \right\} + s_k I \\
 &= \left[\begin{array}{c|c} * & 0 \\ \hline * & \tilde{A}^k - s_k I \end{array} \right] + s_k I \\
 &= \left[\begin{array}{c|c} * & 0 \\ \hline * & s_k \\ \hline * & \tilde{A}^k \end{array} \right] =: \left[\begin{array}{c|c} \bar{A}^{k-1} & 0 \\ \hline * & \tilde{A}^{k-1} \end{array} \right],
 \end{aligned}$$

us die Behauptung folgt.

□

Bemerkung 14

am wenigsten aufwendigsten Rechnungen sind in Schritt 5 durchzuführen. Ist Ausgangssystem jedoch ein Hessenberg-System im Sinne von Abschnitt 3.2.1., so lassen sich die Rechnungen so organisieren, auch $(\bar{A}^k, \bar{B}^k)$ für $k < n$ Hessenberg-Systeme sind, und in Schritt 5 sind nur trianguläre Gleichungssysteme zu lösen. Auf diese

Weise erhält man direkte Verallgemeinerungen der Algorithmen PZN und PZS auf den multivariablen Fall. Die Details werden hier nicht ausgeführt.

Bemerkung 15

Die durch den Algorithmus PZSM berechnete Rückführung ist i.a. komplex. Der Algorithmus läßt sich jedoch so modifizieren, daß jeweils ein Paar konjugiert komplexer Eigenwerte verschoben wird. Die daraus resultierende Rückführung ist dann reell. Dieses Vorgehen wird hier lediglich für den Fall $s_{n-1} = s_n^H$ illustriert. In diesem Fall bestimmt man zunächst eine komplexe Lösung $\begin{bmatrix} d \\ r \end{bmatrix}$, $d \in \mathbb{C}^m$, $r \in \mathbb{C}^n$ von $[B \mid A - s_n I] \begin{bmatrix} d \\ r \end{bmatrix} = 0$, $r \neq 0$, und bildet dann eine *reelle* Matrix $R \in Gl(n)$ der Form $R = [\tilde{R} \quad \text{Re}(r) \quad \text{Im}(r)]$. Setzt man $F := [0 \mid \text{Re}(d) \mid \text{Im}(d)] R^{-1} \in \mathbb{R}^{m \times n}$, so folgt

$$\begin{aligned} R^{-1}(A+BF)R &= R^{-1}[(A+BF)\tilde{R}, (A+BF)\text{Re}(r), (A+BF)\text{Im}(r)] \\ &= R^{-1}[A\tilde{R}, A \cdot \text{Re}(r) + B \cdot \text{Re}(d), A \cdot \text{Im}(r) + B \cdot \text{Im}(d)] \\ &= R^{-1}[A\tilde{R}, \text{Re}(s_n) \cdot \text{Re}(r) - \text{Im}(s_n) \cdot \text{Im}(r), \text{Re}(s_n) \cdot \text{Im}(r) + \\ &\quad + \text{Im}(s_n) \cdot \text{Re}(r)] \\ &= \left[R^{-1}A\tilde{R} \quad \left. \begin{array}{c} 0 \\ \hline \text{Re}(s_n) \quad \text{Im}(s_n) \\ -\text{Im}(s_n) \quad \text{Re}(s_n) \end{array} \right\} \right] \cdot \underbrace{\quad}_2 \end{aligned}$$

Damit ist durch eine *reelle* Rückführung ein konjugiert-komplexes Paar von Eigenwerten zugewiesen, und das transformierte System $R^{-1}(A+BF)R$ ist auch reell.

Bemerkung 16

Die Algorithmen zur Polverschiebung bei multivariablen Systemen in (Eising 1982; Miminis und Paige 1982a) haben eine gewisse Ähnlichkeit zum Algorithmen PZSM. Es wird dort jedoch von modifizierten Kronecker-Hessenberg-Formen ausgegangen.

4.2.2. En bloc Algorithmen und robuste Polzuweisung

In diesem Abschnitt sollen die Ergebnisse aus Abschnitt 4.1.4. zunächst auf den multivariablen Fall übertragen werden. Das folgende Resultat von Moore (1976) kann als Verallgemeinerung von Lemma 1 angesehen werden.

Lemma 2

Es sei $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$, $s = (s_1, \dots, s_n) \in \mathbb{C}_{\text{sym}}^n$ mit $s_i \neq s_j$ für $i \neq j$ und $r_1, \dots, r_n \in \mathbb{C}^n$. Dann existiert eine Matrix $F \in \mathbb{R}^{m \times n}$, so daß $s_1, \dots, s_n$ die Eigenwerte und $r_1, \dots, r_n$ die dazugehörigen Eigenvektoren von $A + BF$ sind genau dann, wenn die folgenden drei Bedingungen erfüllt sind.

- (i) $r_1, \dots, r_n$ sind linear unabhängig;
- (ii) $r_{2i} = (r_{2i-1}^H)^T$ für $i \in \underline{a} := \underline{a}(s)$; $r_{2a+1}, \dots, r_n \in \mathbb{R}$;
- (iii) es gibt $d_1, \dots, d_n \in \mathbb{C}^m$ mit

$$[B, A - s_i I] \begin{bmatrix} d_i \\ r_i \end{bmatrix} = 0 \quad \text{für } i \in \underline{n}. \quad (8)$$

Falls die Bedingungen (i) - (iii) erfüllt sind und $\text{rang}(B) = m$, so ist F eindeutig bestimmt und gegeben durch

$$F = \tilde{F}R^{-1}, \quad (9)$$

wobei

$$\tilde{F} = [\operatorname{Re}(d_1), \operatorname{Im}(d_1), \dots, \operatorname{Re}(d_{2a-1}), \operatorname{Im}(d_{2a-1}), d_{2a+1}, \dots, d_n]; \quad (10)$$

$$R = [\operatorname{Re}(r_1), \operatorname{Im}(r_1), \dots, \operatorname{Re}(r_{2a-1}), \operatorname{Im}(r_{2a-1}), r_{2a+1}, \dots, r_n]. \quad (11)$$

Man erhält also einen en bloc Algorithmus, wenn man die Gleichungssysteme in (8) derart löst, daß die Bedingungen (i) und (ii) erfüllt sind. Da A und B reell sind, kann man die Bedingung (ii) immer erfüllen. Es ist aber nicht klar, wie die Bedingung (i) à priori sichergestellt werden kann. Man beachte, daß für $m = 1$ (Lemma 1) die Bedingung (i) aus der Erreichbarkeit des Systems folgt.

Es wird daher vorgeschlagen, Optimierungsalgorithmen einzusetzen, die bewirken, daß $\kappa(X) := \|X\|_2 \cdot \|X^{-1}\|_2$ minimiert wird, wobei $X := [r_1, \dots, r_n]$. Dieses hat dann den Vorteil, daß die Gleichungssysteme (9) zuverlässig lösbar sind, und daß das System angenehme Robustheitseigenschaften besitzt. Insbesondere reagieren die Eigenwerte des geschlossenen Kreises möglichst insensitiv gegenüber Störungen im System (Wilkinson 1965, Abschnitt 2.30), und die Rückführungsverstärkungen bleiben klein (Kautsky, Nichols, van Dooren und Fletcher 1982). Um Optimierungsalgorithmen einsetzen zu können, wird zunächst die Menge aller Vektoren r_i parametrisiert, die (iii) erfüllen. Das folgende einfache Resultat findet sich in (Moore 1976) und (Kautsky und Nichols 1983).

Lemma 3

Es sei $(A, B) \in \Sigma(n, m)$, $r_i \in \mathbb{C}^n$ und $s_i \in \mathbb{C}$.

Ist $Q_i \in U(n+m, \mathbb{C})$ so gewählt, daß

$$[B, A - s_i I] Q_i = \begin{bmatrix} \underbrace{z_i}_n, & \underbrace{0}_m \end{bmatrix},$$

wobei z_i vollen Spaltenrang besitzt, und ist Q_i wie folgt partitioniert

$$Q_i = \begin{bmatrix} \underbrace{S_i}_n & \underbrace{\begin{bmatrix} \bar{S}_i \\ S_i \end{bmatrix}}_m \end{bmatrix} \begin{matrix} m \\ n \end{matrix},$$

so erfüllt r_i die Bedingung (iii) in Lemma 2 genau dann, wenn

$$r_i \in \text{im } S_i.$$

Falls $r_i = S_i w_i$ mit $w_i \in \mathbb{C}^m$, so ist d_i gegeben durch $d_i = \bar{S}_i w_i$.

Bemerkung 17

Falls (A, B) ein Hessenberg-System ist, so läßt sich die Matrix Q_i besonders einfach bestimmen.

Da man anstatt von κ auch κ^2 minimieren kann, ist im Hinblick auf die Lemmata 2 und 3 das folgende *unbeschränkte* Minimierungsproblem zu lösen.

$$\begin{aligned}
 & \left\{ \begin{array}{l} \text{minimiere } \kappa^2(X), \text{ wobei} \\ X: = [S_1 w_1, ((S_1 w_1)^H)^T, \dots, S_{2a-1} w_{2a-1}, ((S_{2a-1} w_{2a-1})^H)^T, \\ S_{2a+1} w_{2a+1}, \dots, S_n w_n) \\ \text{und } w: = (w_1^T, w_3^T, \dots, w_{2a-1}^T, w_{2a+1}^T, \dots, w_n^T)^T \end{array} \right. .
 \end{aligned}$$

Man beachte, daß $\kappa^2(X) < \infty$ impliziert, daß eine Lösung des Polzuweisungsproblems gefunden ist. Es bezeichne jetzt $\sigma_1(X), \dots, \sigma_n(X)$ die singulären Werte von X und $v_1(X), \dots, v_n(X)$ ihre Quadrate. Dann sind $v_1(X), \dots, v_n(X)$ die Eigenwerte von $X^H X$ und es gilt

$$\kappa^2(X) = \frac{\max\{v_i(X) \mid i \in \underline{n}\}}{\min\{v_i(X) \mid i \in \underline{n}\}} .$$

Daher ist das unbeschränkte Minimierungsproblem P äquivalent zu dem folgenden Minimierungsproblem mit Nebenbedingungen.

$$\begin{aligned}
 P' : \left\{ \begin{array}{l} \text{minimiere } \frac{u}{v} \text{ bezüglich} \\ u \geq v_i(X), \quad i \in \underline{n}, \\ v \leq v_i(X), \quad i \in \underline{n}, \\ \text{wobei } X \text{ wie in } P \text{ definiert ist.} \end{array} \right.
 \end{aligned}$$

Man beachte, daß die Matrix $X^H X$ lineare Elementarteiler besitzt. Falls also $v_1(X), \dots, v_n(X)$ paarweise verschieden sind, so sind die v_i nach (Wilkinson 1965, Abschnitte 2.5-2.9)

differenzierbar und die Ableitungen $\frac{\partial v_i(X)}{\partial (\operatorname{Re} w_k)} , \frac{\partial v_i(X)}{\partial (\operatorname{Im} w_k)}$

lassen sich explizit berechnen. Daher können Algorithmen für

reelle, differenzierbare Minimierungsprobleme mit Nebenbedingungen zur Lösung von P' herangezogen werden. Es kann jedoch zu Schwierigkeiten kommen, wenn der Algorithmus gegen eine Matrix X konvergiert, die entweder singulär ist oder keine einfachen singulären Werte besitzt. Ein Ausweg in einer solchen Situation könnte sein, für einige Iterationen den Güteindex zu ändern. Beispielsweise könnte man den in (Cavin und Bhattacharyya 1982) vorgeschlagenen Güteindex

$$J(X) := \text{spur}[I - X^H X]^2 \quad (12)$$

heranziehen. Es scheint jedoch nicht vorteilhaft zu sein, nur mit dem Güteindex (12) zu arbeiten, da beispielsweise die singuläre Matrix $\tilde{X} := \text{diag}(1, \dots, 1, 0)$ nur $J(\tilde{X}) = 1$ besitzt, während auch die "fast orthogonale" Matrix

$$\hat{X} := \begin{bmatrix} 1 & \epsilon & & 0 \\ & \ddots & \ddots & \\ & & \ddots & \epsilon \\ 0 & & & 1 \end{bmatrix}$$

mit $2(n-1)\epsilon + n\epsilon^2 = 1$ schon $J(\hat{X}) = 1$ besitzt.

Bemerkung 18

Eine Implementierung des oben vorgeschlagenen en bloc Algorithmus steht noch aus.

Bemerkung 19

Die folgenden Arbeiten befassen sich ebenfalls mit der Berechnung einer "robusten Zustandsrückführung".

- (i) *Cavin und Bhattacharyya (1982)*
- (ii) *Kautsky, Nichols, van Dooren und Fletcher (1982),
Kautsky und Nichols (1983).*

Das Verfahren in (i) unterscheidet sich von dem hier vorgeschlagenen Verfahren insofern, als daß in (i) die Pole mit Hilfe der Sylvester-Gleichung zugewiesen werden, und daß das Gütekriterium (12) verwendet wird. In (ii) wird ebenfalls die Parametrisierung aus Lemma 3 verwendet. Allerdings werden dann nur Approximationen von $\kappa(X)$ minimiert.

4.3. Großsysteme mit dünn besetzten Systemmatrizen

Dieser Abschnitt fällt etwas aus dem Rahmen der restlichen Arbeit, da hier eine andere Systemklasse untersucht wird. Das System, gegeben durch $(A,B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$, habe jetzt Struktur in dem Sinne, daß die Matrizen A und B dünn besetzt sind. Prinzipiell sind die Algorithmen der vorherigen Abschnitte auch auf diese Systeme anwendbar. Die Erfahrungen aus der numerischen linearen Algebra zeigen jedoch, daß spezielle Algorithmen, die die Struktur ausnutzen und erhalten, oftmals viel effizienter arbeiten, und so auch auf höherdimensionale Probleme anwendbar sind. Berechnet man beispielsweise eine System-Hessenberg-Form eines Systems mit Struktur, so wird das entstehende System - außer der Hessenberg-Struktur - i.a. keine weitere Struktur besitzen. Die Struktur des Systems wird also zerstört. Dieses Beispiel zeigt, daß die *Zweischrittmethode* für Systeme mit Struktur nicht geeignet ist.

Bei großen Systemen werden i.a. schon viele Eigenwerte stabil sein, und diese sollten - um den Aufwand zu begrenzen - nicht weiter verschoben werden. Es wird hier also das *Stabilisierungsproblem* behandelt. Es werden *zeitdiskrete Systeme* untersucht, bei denen verlangt wird, daß die Eigenwerte in einen Kreis (r fest)

$$K_r := \{s \in \mathbb{C} \mid |s| \leq r\}$$

verschoben werden. Falls $s \in K_r$, so heißt s stabil. Um die Idee des Algorithmus besser herausarbeiten zu können, wird zusätzlich angenommen, daß die instabilen Eigenwerte reell und paarweise verschieden sind. Dann hat der Algorithmus PZDB (Polzuweisung für dünn besetzte Systeme) die folgende Form.

Algorithmus PZDB

1. Eingabe $n, m \geq 1$, $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$.
2. $k := 0$.
3. Berechne den dominanten (betragsgrößten) Eigenwert $s_{k+1} \in \mathbb{R}$ und den dazugehörigen linken Eigenvektor $r_{k+1} \in \mathbb{R}^{1 \times n}$ von $A + B(F_1 R_1 + \dots + F_k R_k)$. [Falls $k = 0$, so tritt $B(F_1 R_1 + \dots + F_k R_k)$ nicht auf.]
4. Falls s_{k+1} stabil ist, so stoppe ($F := F_1 R_1 + \dots + F_k R_k$ ist eine stabilisierende Rückführung).
5. Bilde eine invertierbare Matrix R_{k+1} mit r_{k+1} als letzte Zeile.

6. Berechne $t_{k+1} := r_{k+1} B \in \mathbb{R}^{1 \times m}$.

7. Falls $t_{k+1} = 0$, so stoppe (der Eigenwert s_{k+1} ist instabil und nicht verschiebbar: siehe auch anschließender Beweis).

8. Berechne $f_{k+1} \in \mathbb{R}^m$, so daß $\tilde{s}_{k+1} := s_{k+1} + t_{k+1} f_{k+1}$ stabil ist und setze

$$F_{k+1} := [0, f_{k+1}] \in \mathbb{R}^{m \times n}.$$

9. $k \leftarrow k+1$; gehe zu 3.

Die Berechtigung des Algorithmus PZDB ergibt sich aus der folgenden Überlegung: Aufgrund der Definitionen von r_{k+1} und s_{k+1} gilt

$$r_{k+1}^T A_k = r_{k+1}^T s_{k+1},$$

wobei $A_k := A + B(F_1 R_1 + \dots + F_k R_k)$.

Hieraus folgt

$$R_{k+1} A_k R_{k+1}^{-1} = \begin{bmatrix} * \\ \vdots \\ 0 \dots 0 \ s_{k+1} \end{bmatrix}.$$

und

$$R_{k+1} B = \begin{bmatrix} * \\ \vdots \\ t_{k+1} \end{bmatrix}.$$

Aus dieser Darstellung ergibt sich die Behauptung aus Schritt 7.

Ferner folgt, daß die Rückführung $F_{k+1} R_{k+1}$ den Eigenwert

s_{k+1} nach $\tilde{s}_{k+1}$ verschiebt, während alle anderen Eigenwerte unverändert bleiben. Insbesondere stoppt der Algorithmus spätestens mit $k = n$.

Die aufwendigsten Rechnungen im Algorithmus PZDB sind in Schritt 3 durchzuführen. Es existieren jedoch sehr effiziente Methoden, dominante Eigenwerte und die dazugehörigen Eigenvektoren von großen, dünn besetzten Matrizen zu berechnen. (Man beachte, daß alle auftretenden Matrizen A , B , R_i und F_i und in der Regel auch $A+B(F_1R_1+\dots+F_kR_k)$ dünn besetzt sind). Diese Algorithmen sind Varianten der *direkten Vektor Iteration* (power method) und der *inversen Iteration*. Details findet man beispielsweise in (Ruhe 1977) und (Wilkinson 1965, Kapitel 9).

Der Algorithmus PZDB ist sehr ähnlich zu der von Hotelling vorgeschlagenen Methode der Deflation zur Eigenwertberechnung (Wilkinson 1965, Abschnitt 9.19), hat jedoch den Vorteil, daß keine Ähnlichkeitstransformationen erforderlich sind. Ferner besteht eine gewisse Ähnlichkeit zu dem Polzuweisungsalgorithmus in (Varga 1981a). Die dort vorgeschlagene Methode ist jedoch nicht für dünn besetzte Systeme geeignet, da der QR-Algorithmus eingesetzt wird.

Besitzt ein System mehrere dominante instabile Eigenwerte (etwa ein Paar konjugiert komplexer Zahlen) oder einen "Haufen" von dicht zusammenliegenden dominanten instabilen Eigenwerten, so kann man den Algorithmus PZDB so modifizieren, daß in

Schritt 3 jeweils mehrere Eigenwerte simultan berechnet werden. Insgesamt läßt sich sagen, daß sich die meisten "Tricks" die zur Verbesserung der Eigenschaften der inversen und direkten Iteration erfunden wurden, auch in den Algorithmus PZDB einarbeiten lassen.

Um einen Algorithmus PZDB für *zeitkontinuierliche Systeme* zu bekommen, müßten die Standardverfahren zur Eigenwertberechnung so modifiziert werden, daß sie jeweils die Eigenwerte mit größtem Realteil liefern. Im Prinzip ist dieses möglich, indem man (beispielsweise bei der direkten Iteration) die Folge $(A^k x_0)_{k \in \mathbb{N}}$ ersetzt durch $(\sum_{j=1}^k \frac{1}{j!} (K \cdot A)^j x_0)_{k \in \mathbb{N}}$ mit $K \rightarrow \infty$. Da hierbei aber zwei Grenzprozesse durchzuführen sind, muß sich die praktische Durchführbarkeit dieser Erweiterung erst erweisen.

4.4. Anmerkungen

Die vorhandene Literatur zur numerischen Lösung des Polzuweisungsproblems durch Zustandsrückführung läßt sich wie folgt klassifizieren.

- (i) Algorithmen, die auf Hessenberg-Formen aufbauen:
Nour Eldin (1974) [$m = 1$]; Eising (1982); Miminis und Paige (1982) [$m = 1$]; Miminis und Paige (1982a); Petkov, Konstantinov und Christov (1982) [$m = 1$]; Kattí (1983).

(ii) Andere Zweischrittmethoden:

Sima (1981); *Varga* (1981a); *Bhattacharyya* und *Souza* (1982).

(iii) Keine Zweischrittmethode:

Konstantinov, Petkov und *Christov* (1981a) [$m = 1$].

(iv) Algorithmen für Deadbeat-Control (spezielle Polzuweisung):

Eising (1983), *van Dooren* (1984).

(v) Polzuweisung unter Berücksichtigung von Robustheitsaspekten:

Cavin und *Bhattacharyya* (1982); *Kautsky, Nichols*, *van Dooren* und *Fletcher* (1982); *Kautsky* und *Nichols* (1983).

Hierbei werden in *Nour Eldin* (1974) und *Katti* (1983) nicht die Pole direkt, sondern die Koeffizienten charakteristischer Polynome zugewiesen. Das vorliegende Kapitel 4 liefert Beiträge zu (i), (iii), (v) und zu

(vi) Polzuweisung bei Systemen mit Struktur.

Ein weiterhin offenes Problem ist es, numerisch zuverlässige Algorithmen für das Polzuweisungsproblem durch Ausgangsrückführung zu finden. Partielle Resultate hierzu sind vom Verfasser erzielt worden. Jedoch wird bei allen Algorithmen das Problem zunächst auf ein Problem mit einem Eingang oder Ausgang reduziert. Im Hinblick auf Abschnitt 4.1.6. muß die Relevanz dieser Resultate jedoch angezweifelt werden, und auf eine Darstellung wird hier verzichtet.

5. STÖRUNGSENTKOPPLUNG

- 5.1. Geometrische Theorie der Störungsentkopplung 132
 - 5.1.1. Störungsentkopplung durch Zustandsrückführung 132
 - 5.1.2. Störungsentkopplung durch Ausgangsrückführung 132
 - 5.1.3. Störungsentkopplung und Polzuweisung durch Ausgangsrückführung 137
 - 5.1.4. Fast-Störungsentkopplung mit strenger Stabilisierung durch Zustandsrückführung 139
- 5.2. Analyse von Systemen mit Hilfe von Matrix-Algorithmen 142
 - 5.2.1. Verhältnis zwischen den Unterraum-Algorithmen 142
 - 5.2.2. Ein Matrix-Algorithmus 144
 - 5.2.3. Analyse von Systemen mit Algorithmus MAIU 157
 - 5.2.3.1. Störungsentkopplung durch Zustandsrückführung 157
 - 5.2.3.2. Störungsentkopplung durch Ausgangsrückführung 158
 - 5.2.3.3. Störungsentkopplung und Polzuweisung durch Zustandsrückführung 165
 - 5.2.3.4. Fast-Störungsentkopplung mit strenger Stabilisierung durch Zustandsrückführung
- 5.3. Synthese von Systemen mit Hilfe von Matrix-Algorithmen 165
 - 5.3.1. Störungsentkopplung durch Zustandsrückführung 165
 - 5.3.2. Störungsentkopplung durch Ausgangsrückführung 165
 - 5.3.3. Störungsentkopplung mit Polzuweisung durch Zustandsrückführung 172
 - 5.3.4. Fast-Störungsentkopplung 178
- 5.4. Matrizen- und strukturelle Charakterisierungen der Lösbarkeit 202
 - 5.4.1. Störungsentkopplung durch Zustandsrückführung 202
 - 5.4.2. Störungsentkopplung durch Ausgangsrückführung 202
 - 5.4.3. Störungsentkopplung und Polzuweisung durch Zustandsrückführung 215
 - 5.4.4. Fast-Störungsentkopplung 219

In diesem Kapitel werden Störungsentkopplungsprobleme analysiert. Im Unterschied zu den meisten bisherigen Arbeiten dieses Gebietes liegt das Schwergewicht jedoch auf dem numerischen Aspekt, d.h. es sollen numerisch zuverlässige Algorithmen zur Störungsentkopplung entwickelt werden. Der Leitfaden hierzu ist die in der Einleitung beschriebene "Zweischrittmethode". Bei dem vorliegenden Problem muß jedoch zwischen "Analyse" und "Synthese" unterschieden werden.

Will man für ein gegebenes System erfahren, ob ein Störungsentkopplungsproblem lösbar ist, so wird in Abschnitt 5.2 ein Algorithmus beschrieben, mit dem das System numerisch stabil in ein ähnliches System transformiert werden kann, von dem die Lösbarkeit dann ablesbar ist. Für das einfachste Störungsentkopplungsproblem - Störungsentkopplung durch Zustandsrückführung - erhält man so gleichzeitig eine Methode zur Berechnung des Reglers. Für allgemeinere Problemstellungen sind jedoch weitere Algorithmen nötig, und diese werden in Abschnitt 5.3 beschrieben.

Im Gegensatz zu den Abschnitten 5.2 und 5.3 stehen im Abschnitt 5.4 nicht mehr die algorithmischen Aspekte im Vordergrund. Hier wird gezeigt, daß die speziellen Systeme, die in den Abschnitten 5.2 und 5.3 produziert werden, typisch für bestimmte Störungsentkopplungsprobleme sind, in dem Sinne, daß sich mit deren Hilfe die Lösbarkeit

charakterisieren läßt. Ein typisches Resultat aus Abschnitt 5.4 hat die folgende Form: Das Störungsentkopplungsproblem ... ist genau dann lösbar, wenn sich das System per Ähnlichkeit in die Form ... transformieren läßt. Ein derartiges Resultat wird "matrizielle Charakterisierung" genannt. Übersetzt man solch eine matrizielle Charakterisierung in eine Bedingung an das Signalflußdiagramm, so erhält man eine "strukturelle Charakterisierung". Zunächst jedoch - in Abschnitt 5.1 - werden einige bereits bekannte Ergebnisse der geometrischen Theorie rekapituliert.

In diesem Kapitel werden nicht alle möglichen Versionen des Störungsentkopplungsproblems systematisch abgearbeitet. Es ist vielmehr das Ziel dieses Abschnittes, an ausgewählten Problemen die wesentlichen Ideen zu vermitteln und die auftretenden Schwierigkeiten zu verdeutlichen.

5.1. Geometrische Theorie der Störungsentkopplung

In diesem Abschnitt werden einige Ergebnisse aus der geometrischen Theorie kurz wiedergegeben. Die Ausführungen sind nicht als Einstieg in die Problemstellung und deren Lösungsmethoden geeignet. Vielmehr soll der bereits mit der Materie vertraute Leser an die wesentlichen Definitionen erinnert werden, und es sollen Notationen bereitgestellt werden. Als generelle Literaturquellen für diesen Abschnitt haben (Wonham 1974; Willems 1981 ; Willems 1982; Willems und Commault 1981) gedient.

5.1.1. Störungsentkopplung durch Zustandsrückführung

Gegeben sei ein System der Form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + Gd(t) \\ z(t) &= Hx(t),\end{aligned}\tag{1}$$

wobei $x(t) \in \mathbb{R}^n$ den Zustand, $u(t) \in \mathbb{R}^m$ die Steuerung, $d(t) \in \mathbb{R}^q$ die Störung und $z(t) \in \mathbb{R}^l$ den zu steuernden Ausgang bezeichnet. Das Störungsentkopplungsproblem durch Zustandsrückführung (SEPZ) bedeutet, eine statische Zustandsrückführung

$$u(t) = Fx(t), \quad F \in \mathbb{R}^{m \times n},\tag{2}$$

zu finden, so daß im geschlossenen System Signale $d(\cdot)$ den Ausgang $z(\cdot)$ nicht beeinflussen können:

SEPZ: Gegeben sei (1). Finde (falls möglich) (2) so, daß

$$H(sI - (A+BF))^{-1}G = 0.\tag{3}$$

Die Lösbarkeit dieses Problems läßt sich mit Hilfe von (A,B) -invarianten Unterräumen charakterisieren, die wie folgt definiert sind:

Ein Unterraum $V \subset \mathbb{R}^n$ heißt (A,B) -invariant, falls $AV \subset V + \text{im } B$. Ein Raum ist (A,B) -invariant genau dann, wenn eine Matrix $F \in \mathbb{R}^{m \times n}$ existiert mit $(A+BF)V \subset V$.

Es sei L ein Unterraum von $\mathbb{R}^n$. Dann bezeichnet V_L^* den (eindeutig bestimmten) maximalen (A,B) -invarianten Unterraum in L , d.h. V_L^* ist (A,B) -invariant, in L enthalten und, falls $V \subset L$ ebenfalls (A,B) -invariant ist, folgt $V \subset V_L^*$.

Satz 1

SEPZ ist lösbar genau dann, wenn

$$\text{im } G \subset V_{\ker H}^*.$$

Der Unterraum V_L^* ist zentral in der geometrischen Theorie und läßt sich durch den folgenden Unterraum-Algorithmus berechnen:

Algorithmus 1

$$V_L^0 := \mathbb{R}^n$$

$$V_L^{k+1} := L \cap A^{-1}(V_L^k + \text{im } B).$$

Für die durch Algorithmus 1 berechnete Folge $V_L^0, V_L^1, V_L^2, \dots$ von Unterräumen des $\mathbb{R}^n$ gilt

$$V_L^0 \supset V_L^1 \supset V_L^2 \supset \dots,$$

$$V_L^{k+1} = V_L^k \Rightarrow V_L^{k+i} = V_L^k = V_L^\infty := \lim_{k \rightarrow \infty} V_L^k \quad \text{für alle } i \geq 1,$$

$$V_L^{(\dim L)+1} = V_L^\infty,$$

$$V_L^\infty = V_L^*.$$

5.1.2 Störungsentkopplung durch Ausgangsrückführung

Zusätzlich zu (1) sei noch die Beziehung

$$y(t) = Cx(t) \quad (4)$$

gegeben, wobei $y(t) \in \mathbb{R}^D$ den gemessenen Ausgang bezeichnet. Betrachtet wird dynamische Ausgangsrückführung der Form

$$\begin{aligned} u(t) &= Mw(t) + Ny(t) \\ \dot{w}(t) &= Fw(t) + Ey(t), \end{aligned} \quad (5)$$

wobei $w(t) \in \mathbb{R}^S$ den Zustand des Reglers bezeichnet.

Das geschlossene System hat dann die Form

$$\begin{aligned} \dot{x}_C(t) &= A_C x_C(t) + G_C d(t) \\ z(t) &= H_C x_C(t), \end{aligned} \quad (6)$$

wobei

$$\begin{aligned} x_C &= \begin{bmatrix} x \\ w \end{bmatrix}, & A_C &= \begin{bmatrix} A+BNC & BM \\ EC & F \end{bmatrix}, \\ G_C &= \begin{bmatrix} G \\ 0 \end{bmatrix}, & H_C &= \begin{bmatrix} H & 0 \end{bmatrix}. \end{aligned} \quad (7)$$

Das Störungsentkopplungsproblem durch Ausgangsrückführung lautet

SEPA: Gegeben sei (1) und (4). Finde (falls möglich)

(5) so, daß

$$H_C (sI - A_C)^{-1} G_C = 0. \quad (8)$$

Zur Charakterisierung der Lösbarkeit dieses Problems wird das duale Konzept zum maximalen (A,B) -invarianten Unterraum benötigt: Ein Unterraum $W \subset \mathbb{R}^n$ heißt (C,A) -invariant falls $A(W \cap \ker C) \subset W$. Es sei jetzt L ein Unterraum des $\mathbb{R}^n$. Dann bezeichnet W_L^* den minimalen (C,A) -invarianten L enthaltenden Unterraum, d.h. W_L^* ist (C,A) -invariant, $L \subset W_L^*$, und W_L^* ist minimal bezüglich dieser Eigenschaften.

Satz 2

SEPA ist lösbar genau dann, wenn

$$W_{\text{im}G}^* \subset V_{\ker H}^*$$

Durch Dualisierung von Algorithmus 1 läßt sich W_L^* berechnen:

Algorithmus 2

$$W_L^0 = \{0\}$$

$$W_L^{k+1} = L + A(W_L^k \cap \ker C).$$

Analog zum Algorithmus 1 gilt für die durch Algorithmus 2 berechnete Folge $W_L^0, W_L^1, W_L^2, \dots$ von Unterräumen:

$$W_L^0 \subset W_L^1 \subset W_L^2 \subset \dots,$$

$$W_L^{k+1} = W_L^k \Rightarrow W_L^{k+i} = W_L^k = W_L^\infty = \lim_{k \rightarrow \infty} W_L^k \text{ für alle } i \geq 1,$$

$$W_L^{n-(\dim L)+1} = W_L^\infty,$$

$$W_L^\infty = W_L^*.$$

5.1.3 Störungsentkopplung und Polzuweisung durch Zustandsrückführung

Wenn eine Zustandsrückführung zusätzlich zu (3) dem System noch vorgegebene Pole zuweisen soll, so spricht man vom Störungsentkopplungsproblem mit Polzuweisung durch Zustandsrückführung:

SEPPZ Gegeben sei (1). Finde (falls möglich) zu jeder Menge $C_g \subset \mathbb{C}$ (abgeschlossen unter Konjugation; $C_g \cap \mathbb{R} \neq \emptyset$) einen Regler (2) so, daß (3) erfüllt ist und $\text{spec}(A+BF) \subset C_g$.

Die Lösbarkeit dieses Problems läßt sich mit Kontrollierbarkeitsräumen charakterisieren, die wie folgt definiert sind.

Ein Unterraum R des $\mathbb{R}^n$ heißt Kontrollierbarkeitsunterraum, falls es $F \in \mathbb{R}^{m \times n}$ und einen Unterraum $U \subset \text{im } B$ gibt mit

$$R = \langle A+BF | U \rangle,$$

wobei $\langle A+BF | U \rangle := U + (A+BF)U + \dots + (A+BF)^{n-1}U$.

Es sei L ein Unterraum des $\mathbb{R}^n$. Dann bezeichnet R_L^* den maximalen Kontrollierbarkeitsunterraum in L , d.h. $R_L^* \subset L$ ist Kontrollierbarkeitsunterraum und jeder andere Kontrollierbarkeitsunterraum $R \subset L$ ist in R_L^* enthalten.

Satz 3

SEPPZ ist lösbar genau dann, wenn (A,B) erreichbar ist und

$$\text{im } G \subset R_{\ker H}^*$$

Zur Berechnung von R_L^* gibt es mindestens zwei verschiedene Methoden, die beide den maximalen (A,B) -invarianten Unterraum in L benutzen.

Satz 4

Es sei $F \in \mathbb{R}^{m \times n}$ so gewählt, daß $(A+BF)V_L^* \subset V_L^*$.

Dann gilt

$$R_L^* = \langle A+BF | \text{im } B \cap V_L^* \rangle.$$

Die zweite Methode benutzt den folgenden Unterraumalgorithmus für einen Unterraum $L \subset \mathbb{R}^n$.

Algorithmus 3

$$R_L^0 = \{0\}$$

$$R_L^{k+1} = L \cap (AR^k + \text{im } B)$$

Für die Folge $R_L^0, R_L^1, R_L^2, \dots$ gilt

$$R_L^0 \subset R_L^1 \subset R_L^2 \subset \dots,$$

$$R_L^{k+1} = R_L^k \Rightarrow R_L^{k+1} = R_L^k = R_L^\infty = \lim_{k \rightarrow \infty} R_L^k \text{ für alle } i \geq 1,$$

$$R_L^{\dim L} = R_L^\infty,$$

$$R_L^* = V_L^\infty \cap R_L^\infty = R_{V_L^\infty}^\infty = V_{R_L^\infty}^\infty.$$

Wird das zu SEPPZ duale Problem ("disturbance decoupled estimation problem") untersucht, so führt dieses auf den dualen Algorithmus zu Algorithmus 3:

Algorithmus 4

$$S_L^0 : = \mathbb{R}^n$$

$$S_L^{k+1} : = L + (A^{-1} S_L^k \cap \ker C).$$

5.1.4 Fast-Störungsentkopplung mit strenger Stabilisierung durch Zustandsrückführung

Fordert man nicht, wie in SEPPZ, daß die Transferfunktion von d nach z verschwindet, sondern daß sie beliebig klein wird, so erhält man das Fast-Störungsentkopplungsproblem mit strenger Stabilisierung durch Zustandsrückführung. Es sei $1 \leq p \leq \infty$.

FSEPSSZ_p: Gegeben sei (1). Für alle $\varepsilon > 0$ und alle $M \in \mathbb{R}$ finde (falls möglich) einen Regler (2), so daß für die Lösung von (1) und (2) mit $x(0) = 0$ gilt

$$\|z\|_{L_p(0,\infty)} \leq \varepsilon \|d\|_{L_p(0,\infty)},$$

und für die Eigenwerte $s_1, \dots, s_n$ von $A+BF$ gilt

$$\operatorname{Re}(s_i) \leq M, \quad i \in \underline{n}.$$

Satz 5

Es sei $1 \leq p \leq \infty$. FSEPSSZ_p ist lösbar genau dann, wenn (A,B) erreichbar ist und

$$\text{im } G \subset R_{D, \ker H}^*$$

Hierbei bezeichnet $R_{D, \ker H}^*$ den in (Willems 1981) definierten fast-(A,B)-invarianten Unterraum, für den

$$R_{D, L}^* = AR_L^\infty + \text{im } B$$

gilt. Damit kann $R_{D, L}^*$ durch Algorithmus 3 berechnet werden. Ein direkterer Algorithmus ist

Algorithmus 5

$$W_L^0 = \{0\}$$

$$W_L^{k+1} = \text{im } B + A(L \cap W_L^k).$$

Es gilt nämlich für die Folge $W_L^0, W_L^1, W_L^2, \dots$ aus Algorithmus 5:

$$R_{D, L}^* = W_L^\infty = \lim_{k \rightarrow \infty} W_L^k$$

Man beachte, daß die Charakterisierung in Satz 5 unabhängig von p ist. Man kann also den Index p bei FSEPSSZ_p fortlassen. Das zu FSEPSSZ duale Problem führt auf den dualen Algorithmus zu Algorithmus 5:

Algorithmus 6

$$V_L^0 = \mathbb{R}^n$$

$$V_L^{k+1} = \ker C \cap A^{-1}(V_L^k + L).$$

5.2. Analyse von Systemen mit Hilfe von Matrix-Algorithmen

Der vorherige Abschnitt 5.1. hat gezeigt, daß gewissen Unterraum-Algorithmen bei der Analyse von Systemen, speziell bei der Überprüfung der Lösbarkeit von Störungsentkopplungsproblemen eine wichtige Rolle spielen. Es zeigt sich sogar, daß sich mit Hilfe der Algorithmen 1 - 6 die Lösbarkeit aller in (Willems 1981; Willems 1982; Willems und Commault 1981) behandelten Störungsentkopplungsprobleme überprüfen läßt, und daß sich die Algorithmen auch zur Analyse weiterer Systemeigenschaften heranziehen lassen (vgl. Wonham (1974), Theorem 4.4, Theorem 7.3, Theorem 9.1, etc.). Im Prinzip können diese Algorithmen natürlich direkt mit Hilfe des in (Klema und Laub 1980) beschriebenen "calculus of subspaces" auf einer Maschine implementiert werden. Dieses hat jedoch den Nachteil, daß dann die spezielle Struktur der zu verknüpfenden Unterräume nicht ausgenutzt wird ($L = \ker H$ muß in Algorithmus 1 beispielsweise $(\dim L)$ mal berechnet werden!). Die Folge ist erhöhter Rechenaufwand und damit verbunden eine Verschlechterung der Genauigkeit des Ergebnisses. Weiterhin spricht gegen dieses Vorgehen die Tatsache, daß man primär an den das System beschreibenden Matrizen interessiert ist, und nicht an gewissen Unterräumen (bzw. dessen Basen). Beispielsweise läßt sich der Algorithmus SHF aus Kapitel 3 interpretieren als eine Matrix-Version des Unterraum-Algorithmus

$$\begin{aligned} w^0 &= \{0\} \\ w^{k+1} &= \operatorname{im} B + A w^k. \end{aligned} \tag{9}$$

Man war jedoch nicht primär an den Räumen W^k , sondern an der Struktur der transformierten Matrizen interessiert. In diesem Abschnitt werden Matrix-Versionen der Unterraum-Algorithmen aus Abschnitt 5.1 zur Analyse von Systemen herangezogen. Der Algorithmus SHF wird dabei als Spezialfall auftreten (man beachte, daß (9) ein Spezialfall von Algorithmus 2 ist).

Für den Algorithmus 1 ist ein derartiger Matrix-Algorithmus bereits bekannt (Aplevich 1979; van Dooren 1981). Dieser wird in Abschnitt 5.2.2 beschrieben. In Abschnitt 5.2.1 wird jedoch gezeigt, daß ein sehr enger Zusammenhang zwischen den Algorithmen 1 - 6 besteht, und es folgt, daß der Algorithmus aus Abschnitt 5.2.2 universell zur Analyse von Systemen eingesetzt werden kann; vgl. Abschnitt 5.2.3.

5.2.1 Verhältnis zwischen den Unterraum-Algorithmen

In diesem Abschnitt wird der Zusammenhang zwischen den sechs Algorithmen aus Abschnitt 5.1 untersucht. Zusammenfassend treten die folgenden Algorithmen auf, wobei $K \subset \mathbb{R}^n$ stets das Bild einer Matrix, $L \subset \mathbb{R}^n$ stets den Kern einer Matrix bezeichnet, und $A \in \mathbb{R}^{n \times n}$ ist.

Unterraum-Algorithmus UA1

$$V^0(A, K, L): \quad = \mathbb{R}^n$$

$$V^{k+1}(A, K, L): \quad = L \cap A^{-1}(V^k(A, K, L) + K)$$

Unterraum-Algorithmus UA2

$$W^0(A, K, L): = \{0\}$$

$$W^{k+1}(A, K, L): = K + A(W^k(A, K, L) \cap L)$$

Unterraum-Algorithmus UA3

$$R^0(A, K, L): = \{0\}$$

$$R^{k+1}(A, K, L): = L \cap (AR^k(A, K, L) + K)$$

Unterraum-Algorithmus UA4

$$S^0(A, K, L): = \mathbb{R}^n$$

$$S^{k+1}(A, K, L): = K + (A^{-1}S^k(A, K, L) \cap L)$$

Die folgende Proposition ist bereits bekannt.

Proposition 1

Für $k \geq 0$ und $k = \infty$ gilt

$$(i) \quad W^k(A, K, L) = [V^k(A^T, L^\perp, K^\perp)]^\perp,$$

$$(ii) \quad R^k(A, K, L) = W^k(A, K, L) \cap L = [V^k(A^T, L^\perp, K^\perp) + L^\perp]^\perp,$$

$$(iii) \quad S^k(A, K, L) = [R^k(A^T, L^\perp, K^\perp)]^\perp = V^k(A, K, L) + L.$$

Beweis

Der Beweis wird durch vollständige Induktion nach k geführt. Für $k = 0$ sind die Behauptungen richtig.

Es seien die Behauptungen richtig für ein festes $k \geq 0$.

Dann gilt mit $V^k: = V^k(A, K, L)$, $W^k: = W^k(A, K, L)$,

$R^k: = R^k(A, K, L)$, $S^k: = S^k(A, K, L)$:

$$(i) \quad [W^{k+1}]^\perp = [K + A(W^k \cap L)]^\perp = K^\perp \cap (A^T)^{-1}([W^k]^\perp + L^\perp) \\ = V^{k+1}(A^T, L^\perp, K^\perp)$$

$$(ii) \quad R^{k+1} = L \cap (AR^k + K) = L \cap (A(W^k \cap L) + K) = L \cap W^{k+1}$$

$$(iii) \quad [S^{k+1}]^\perp = [K + (A^{-1}S^k \cap L)]^\perp = K^\perp \cap (A^T[S^k]^\perp + L^\perp) \\ = R^{k+1}(A^T, L^\perp, K^\perp).$$

□

Proposition 1 zeigt, daß die Algorithmen 1 - 6 sämtlich auf den Algorithmus UA1 zurückgeführt werden können.

5.2.2 Ein Matrix-Algorithmus

In diesem Abschnitt wird eine Matrix-Version des Algorithmus UA1 zusammen mit seinen Eigenschaften vorgestellt. Die Resultate stammen von *Aplevich* (1979) und *van Dooren* (1981).

Zunächst sei bemerkt, daß Ähnlichkeitstransformationen zulässig sind:

Lemma 1

Es sei $K = \text{im } B$, $L = \ker H$, $k \geq 0$ oder $k = \infty$, und $S \in \text{Gl}(n)$. Dann gilt

$$V^k(A, \text{im } B, \ker H) = SV^k(S^{-1}AS, \text{im}(S^{-1}B), \ker(HS)). \quad (10)$$

Beweis

Für $k = 0$ stimmt die Behauptung. Es sei jetzt (10) richtig für ein festes $k \geq 0$ und $V^k = V^k(A, \text{im } B, \ker H)$. Dann gilt auch

$$\begin{aligned}
 S^{-1}V^{k+1} &= S^{-1}(\ker H) \cap S^{-1}(A^{-1}(V^k + \operatorname{im} B)) \\
 &= \ker(HS) \cap (S^{-1}AS)^{-1}(S^{-1}V^k + \operatorname{im}(S^{-1}B)) \\
 &= \ker(HS) \cap (S^{-1}AS)^{-1}(V^k(S^{-1}AS, \operatorname{im}(S^{-1}B), \ker(HS)) + \operatorname{im}(S^{-1}B)) \\
 &= V^{k+1}(S^{-1}AS, \operatorname{im}(S^{-1}B), \ker(HS)).
 \end{aligned}$$

□

Der folgende Algorithmus MAIU (Matrix-Algorithmus zur Berechnung von invarianten Unterräumen) führt orthogonale Ähnlichkeitstransformationen an A, B und H aus.

Algorithmus MAIU

1. Eingabe $n, m, \ell \geq 1$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $H \in \mathbb{R}^{\ell \times n}$

2. Schiebe die Spalten von H nach rechts mit $Q^1 \in O(n)$:

$$HQ^1 = \begin{bmatrix} \underbrace{O}_{:t_1} & \widetilde{H}^0 \end{bmatrix},$$

wobei $\widetilde{H}^0$ vollen Spaltenrang besitzt.

3. Berechne

$$A^1 := (Q^1)^T A Q^1$$

$$B^1 := (Q^1)^T B$$

$$H^1 := HQ^1$$

4. Falls $t_1 = n$, d.h. $H = O$, so setze

$$r_1 := n, \quad r_2 := 0, \quad r_3 := 0$$

$$\hat{A} = A^1, \quad \hat{B} = B^1, \quad \hat{H} = H^1, \quad Q = Q^1$$

und stoppe.

5. Falls $t_1 = 0$, d.h. H vollen Spaltenrang besitzt,
so setze

$$r_1 = 0, \quad r_2 = n, \quad r_3 = 0$$

$$\hat{A} = A^1, \quad \hat{B} = B^1, \quad \hat{H} = H^1, \quad Q = Q^1$$

und stoppe.

6. $k := 1, \quad s_1 := 0$

7. Kommentar: Es gilt (vgl. anschließenden Beweis)

$$(i) \quad A^k = (Q^k)^T A Q^k$$

$$B^k = (Q^k)^T B$$

$$H^k = H Q^k$$

$$Q^k \in O(n)$$

$$(ii) \quad 0 < t_k < n$$

$$0 \leq s_k \leq n$$

$$0 < n - t_k - s_k \leq n$$

(iii) A^k, B^k, H^k haben die Struktur

$$A^k = \begin{array}{|c|c|c|} \hline \overbrace{\phantom{A^k_{11}}}^{t_k} & \overbrace{\phantom{A^k_{12}}}^{s_k} & \\ \hline A^k_{11} & A^k_{12} & A^k_{13} \\ \hline A^k_{21} & A^k_{22} & A^k_{23} \\ \hline 0 & A^k_{32} & A^k_{33} \\ \hline \end{array} \begin{array}{l} \left. \begin{array}{c} \\ \\ \end{array} \right\} t_k \\ \left. \begin{array}{c} \\ \end{array} \right\} s_k \end{array}$$

$$B^k = \begin{array}{|c|} \hline \overbrace{}^m \\ \hline B^k_1 \\ \hline B^k_2 \\ \hline 0 \\ \hline \end{array} \begin{array}{l} \left. \begin{array}{c} \\ \end{array} \right\} t_k \\ \left. \begin{array}{c} \\ \end{array} \right\} s_k \end{array}$$

$$H^k = \begin{array}{|c|c|c|} \hline 0 & H^k_2 & H^k_3 \\ \hline \end{array} \begin{array}{l} \left. \begin{array}{c} \\ \end{array} \right\} \ell \\ \left. \begin{array}{c} \end{array} \right\} s_k \end{array}$$

(iv) In (iii) gilt:

$$\operatorname{rg} \begin{bmatrix} A_{32}^k & A_{33}^k - sI \\ H_2^k & H_3^k \end{bmatrix} = n - t_k \quad \forall s \in \mathbb{C}.$$

8. Schiebe die Zeilen von B_2^k nach oben mit $(\tilde{Q}_2^k)^T \in O(n - t_k - s_k)$:

$$(\tilde{Q}_2^k)^T B_2^k = \begin{bmatrix} \tilde{B}_2^k \\ 0 \end{bmatrix} : \tilde{s}_k,$$

wobei $\tilde{B}_2^k$ vollen Zeilenrang besitzt.

9. Bilde

$$\tilde{Q}_2^k := \begin{bmatrix} I & 0 & 0 \\ 0 & \tilde{Q}_2^k & 0 \\ 0 & 0 & I \end{bmatrix} \begin{matrix} \} t_k \\ \\ \} s_k \end{matrix} \in O(n).$$

$t_k \qquad s_k$

10. Berechne

$$\tilde{A}^k := (\tilde{Q}^k)^T A^k \tilde{Q}^k$$

$$\tilde{B}^k := (\tilde{Q}^k)^T B^k$$

$$\tilde{H}^k := H^k \tilde{Q}^k$$

$$\tilde{Q}^k := Q^k \cdot \tilde{Q}^k$$

11. Falls $\tilde{s}_k = 0$, d.h. falls B_2^k vollen Zeilenrang besitzt, so setze

$$r_1 = t_k \qquad \hat{A} = \tilde{A}^k$$

$$r_2 = n - t_k - s_k \qquad \hat{B} = \tilde{B}^k$$

$$r_3 = s_k \qquad \hat{H} = \tilde{H}^k$$

$$Q = \tilde{Q}^k$$

und stoppe.

12. $s_{k+1} := s_k + \tilde{s}_k$

13. Kommentar: $\tilde{A}^k, \tilde{B}^k, \tilde{H}^k$ haben die Struktur

$$\tilde{A}^k = \begin{array}{c} \begin{array}{|c|c|c|} \hline \begin{array}{c} \overbrace{}^{t_k} \\ x \end{array} & x & x \\ \hline x & x & x \\ \hline \tilde{A}_{31}^k & \tilde{A}_{32}^k & \tilde{A}_{33}^k \\ \hline 0 & \tilde{A}_{42}^k & \tilde{A}_{43}^k \\ \hline \end{array} & \begin{array}{l} \overbrace{}^{s_k} \\ \end{array} \\ \left. \begin{array}{l} \phantom{\tilde{A}^k} \\ \phantom{\tilde{A}^k} \\ \phantom{\tilde{A}^k} \\ \phantom{\tilde{A}^k} \end{array} \right\} \begin{array}{l} t_k \\ \\ \tilde{s}_k \\ s_k \end{array} \end{array}$$

$$\tilde{B}^k = \begin{array}{|c|} \hline x \\ \hline \tilde{B}_2^k \\ \hline 0 \\ \hline 0 \\ \hline \end{array} \left\{ \begin{array}{l} t_k \\ \\ \tilde{s}_k \\ s_k \end{array} \right.$$

$$\tilde{H}^k = \begin{array}{|c|c|c|} \hline 0 & \tilde{H}_2^k & \tilde{H}_3^k \\ \hline \end{array}$$

mit $\tilde{s}_k > 0$. $\underbrace{}_{t_k}$ $\underbrace{}_{s_k}$

14. Schiebe die Spalten von $\tilde{A}_{31}^k$ nach rechts mit $\tilde{Q}_1^k \in O(t_k)$:

$$\tilde{A}_{31}^k \tilde{Q}_1^k = \begin{bmatrix} \underbrace{0}_{:t_{k+1}} & \tilde{A}_{31}^k \end{bmatrix}$$

wobei $\tilde{A}_{31}^k$ vollen Spaltenrang besitzt.

15. Bilde

$$\tilde{Q}^k = \begin{bmatrix} \tilde{Q}_1^k & 0 \\ 0 & I \end{bmatrix} \in O(n).$$

16. Berechne

$$A^{k+1} = (\bar{Q}^k)^T \tilde{A}^k \bar{Q}^k$$

$$B^{k+1} = (\bar{Q}^k)^T \tilde{B}^k$$

$$H^{k+1} = \tilde{H}^k \bar{Q}^k$$

$$Q^{k+1} = \tilde{Q}^k \bar{Q}^k$$

17. Falls $t_{k+1} = t_k$, d.h. falls $\tilde{A}_{31}^k = 0$, so setze

$$r_1 = t_k \quad \hat{A} = A^{k+1}$$

$$r_2 = n - t_k - s_{k+1} \quad \hat{B} = B^{k+1}$$

$$r_3 = s_{k+1} \quad \hat{H} = H^{k+1}$$

$$Q = Q^{k+1}$$

und stoppe.

18. Falls $t_{k+1} = 0$, d.h. falls $\tilde{A}_{31}^k$ vollen Spaltenrang besitzt, so setze

$$r_1 = 0 \quad \hat{A} = A^{k+1}$$

$$r_2 = n - s_{k+1} \quad \hat{B} = B^{k+1}$$

$$r_3 = s_{k+1} \quad \hat{H} = H^{k+1}$$

$$Q = Q^{k+1}$$

und stoppe.

19. $k \leftarrow k+1$.

20. Gehe zu 7.

Beweis der Behauptungen in Schritt 7

Die Behauptungen (i) - (iii) rechnet man sehr schnell nach.

Zum Beweis von (iv) sei zunächst bemerkt, daß die Behauptung

für $k = 1$ richtig ist. Die Behauptung sei jetzt richtig für ein bestimmtes $k \geq 1$. Dann wird gezeigt, daß die Behauptung auch richtig ist für $k + 1$ (falls der Algorithmus Schritt 7 für $k + 1$ durchläuft). Es gilt nämlich

$$\begin{bmatrix} A_{32}^{k+1} & A_{33}^{k+1} - sI \\ H_2^{k+1} & H_3^{k+1} \end{bmatrix} = \begin{bmatrix} \underbrace{\bar{A}_{31}^k}_{t_{k+1}-t_k} & * & * \\ 0 & \underbrace{\tilde{A}_{42}^k}_{n-t_k-s_k} & \underbrace{\tilde{A}_{43}^k - sI}_{s_k} \\ 0 & \underbrace{\tilde{H}_2^k} & \underbrace{\tilde{H}_3^k} \end{bmatrix} \begin{matrix} \} \tilde{s}_k \\ \} s_k \\ \} \ell \end{matrix} .$$

Ferner gilt

$$\begin{bmatrix} \tilde{A}_{42}^k & \tilde{A}_{43}^k - sI \\ \tilde{H}_2^k & \tilde{H}_3^k \end{bmatrix} = \begin{bmatrix} A_{32}^k & A_{33}^k - sI \\ H_2^k & H_3^k \end{bmatrix} \cdot \begin{bmatrix} \tilde{Q}_2^k & 0 \\ 0 & I \end{bmatrix} ,$$

d.h. die Matrix

$$\begin{bmatrix} \tilde{A}_{42}^k & \tilde{A}_{43}^k - sI \\ \tilde{H}_2^k & \tilde{H}_3^k \end{bmatrix}$$

hat vollen Spaltenrang für alle $s \in \mathbb{C}$. Da $\bar{A}_{31}^k$ vollen Spaltenrang besitzt und unabhängig von $s \in \mathbb{C}$ ist, folgt die Behauptung. □

Man rechnet schnell nach:

Lemma 2

Für die vom Algorithmus berechneten $\hat{A}, \hat{B}, \hat{A}, Q,$

r_1, r_2, r_3 gilt:

$$(i) \quad \hat{A} = Q^T A Q$$

$$\hat{B} = Q^T B$$

$$\hat{H} = H Q$$

$$Q \in O(n)$$

$$(ii) \quad r_1 \geq 0, \quad r_2 \geq 0, \quad r_3 \geq 0, \quad r_1 + r_2 + r_3 = n$$

(iii) $\hat{A}$, $\hat{B}$ und $\hat{H}$ haben die Struktur

$$\hat{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \\ 0 \end{bmatrix},$$

$$H = [0 \quad H_2 \quad H_3],$$

wobei $A_{ij} \in \mathbb{R}^{r_i \times r_j}$, $B_i \in \mathbb{R}^{r_i \times m}$, $H_i \in \mathbb{R}^{l \times r_i}$.

(iv) Falls $r_1 > 0$ und $r_2 > 0$, so gilt in (iii):

$$\text{rg } B_2 = r_2.$$

(v) Falls $r_2 + r_3 > 0$ so gilt in (iii):

$$\text{rg} \begin{bmatrix} A_{32} & A_{33} - sI \\ H_2 & H_3 \end{bmatrix} = r_2 + r_3 \quad \forall s \in \mathbb{C}.$$

Proposition 2

(i) Bezüglich der in Schritt 7 von Algorithmus MAIU konstruierten Basis wird v^k von den ersten t_k Einheitsvektoren aufgespannt, d.h.

$$v^k(A^k, \text{im } B^k, \ker H^k) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \}_{t_k}.$$

(ii) Bezüglich der vom Algorithmus MAIU konstruierten Basis wird V^∞ von den ersten r_1 Einheitsvektoren aufgespannt, d.h.

$$V^\infty(\hat{A}, \text{im } \hat{B}, \ker \hat{H}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \}_{r_1},$$

$$V^\infty(A, \text{im } B, \ker H) = \text{im } Q_1,$$

$$\text{wobei } Q = \left[\underbrace{Q_1}_{r_1}, \underbrace{Q_2}_{n-r_1} \right].$$

Beweis

zu (i): Für $k = 1$ gilt

$$V^1(A^1, \text{im } B^1, \ker H^1) = \ker H^1 = \ker [O \quad \tilde{H}^0] = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \}_{t_1},$$

also die Behauptung. Die Behauptung sei nun richtig für ein bestimmtes $k \geq 1$, d.h. es gelte

$$V^k = V^k(A^k, \text{im } B^k, \ker H^k) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \}_{t_k}.$$

Für $k + 1$ gilt im Schritt 7:

$$A^{k+1} = \begin{array}{|c|c|} \hline & * \\ \hline 0 & M_2 \\ \hline \end{array} \left. \vphantom{\begin{array}{|c|c|}} \right\} s_k$$

$$B^{k+1} = \begin{array}{|c|} \hline * \\ \hline M_1 \\ \hline 0 \\ \hline \end{array} \left. \vphantom{\begin{array}{|c|}} \right\} \begin{array}{l} t_k \\ s_{k+1} \end{array}$$

$$H^{k+1} = \begin{array}{|c|c|} \hline 0 & M_3 \\ \hline \end{array} \left. \vphantom{\begin{array}{|c|c|}} \right\} t_{k+1}$$

wobei M_1 vollen Zeilenrang und $\begin{bmatrix} M_2 \\ M_3 \end{bmatrix}$ vollen Spaltenrang besitzt. Da

$$Q^{k+1} = Q^k \cdot \begin{bmatrix} I & 0 & 0 \\ 0 & \tilde{Q}_2^k & 0 \\ 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} \bar{Q}_1^k & 0 \\ 0 & I \end{bmatrix}$$

gilt, folgt

$$\begin{aligned} & \begin{bmatrix} I & 0 & 0 \\ 0 & \tilde{Q}_2^k & 0 \\ 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} \bar{Q}_1^k & 0 \\ 0 & I \end{bmatrix} V^k(A^{k+1}, \text{im } B^{k+1}, \ker H^{k+1}) \\ &= V^k(A^k, \text{im } B^k, \ker H^k) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_k, \end{aligned}$$

d.h. es gilt auch

$$V^k(A^{k+1}, \text{im } B^{k+1}, \ker H^{k+1}) = \text{im} \begin{bmatrix} (\bar{Q}_1^k)^T \\ 0 \end{bmatrix} = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_k.$$

Es sei jetzt $x = \begin{bmatrix} x_1 \\ 0 \end{bmatrix} \} t_{k+1} \in \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_{k+1}$. Dann gilt einerseits

$$H^{k+1}x = 0$$

und andererseits

$$A^{k+1}x = \begin{bmatrix} y_1 \\ 0 \end{bmatrix} \} s_{k+1} \in \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_k + \text{im } B^{k+1}.$$

Also insgesamt

$$\begin{aligned} x &\in \ker H^{k+1} \cap (A^{k+1})^{-1} (V^k(A^{k+1}, \text{im } B^{k+1}, \ker H^{k+1}) + \text{im } B^{k+1}) \\ &= V^{k+1}(A^{k+1}, \text{im } B^{k+1}, \ker H^{k+1}). \end{aligned}$$

Andersherum sei

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in V^{k+1}(A^{k+1}, \text{im } B^{k+1}, \ker H^{k+1}).$$

Dann gilt einerseits $H^{k+1}x = 0$, d.h.

$$M_3 x_2 = 0$$

und andererseits

$$A^{k+1}x \in \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_k + \text{im } B^{k+1} = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} s_{k+1},$$

d.h.

$$M_2 x_2 = 0.$$

Da aber $\begin{bmatrix} M_2 \\ M_3 \end{bmatrix}$ vollen Spaltenrang besitzt, folgt $x_2 = 0$,
d.h.

$$x \in \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_{k+1},$$

und (i) ist bewiesen.

zu (ii): Falls der Algorithmus in den Schritten 4 oder 5
abbricht, so ist die Behauptung in (ii) trivial erfüllt.

Falls der Algorithmus in Schritt 11 abbricht, so gilt

$$\begin{aligned} V^{k+1}(A^k, \text{im } B^k, \ker H^k) &= \ker H^k \cap (A^k)^{-1}(\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_k + \text{im } B^k) \\ &= \ker H^k \cap (A^k)^{-1}(\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} s_k) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} t_k \\ &= V^k(A^k, \text{im } B^k, \ker H^k), \end{aligned}$$

d.h.

$$\begin{aligned} V^{\infty}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) &= (\hat{Q}^k)^T V^{\infty}(A^k, \text{im } B^k, \ker H^k) \\ &= (\hat{Q}^k)^T \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t_k = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t_k. \end{aligned}$$

Falls der Algorithmus in den Schritten 17 oder 18 abbricht, so folgt wegen

$$\begin{aligned} V^k(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) &= (\hat{Q}^k \hat{Q}^k)^T V^k(A^k, \text{im } B^k, \ker H^k) \\ &= \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t_k \end{aligned}$$

die Beziehung

$$\begin{aligned} V^{k+1}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) &= (\ker \hat{A}) \cap (\hat{A})^{-1} (\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t_k + \text{im } \hat{B}) \\ &= (\ker \hat{A}) \cap (\hat{A})^{-1} (\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} s_{k+1}). \end{aligned}$$

Falls $t_{k+1} = t_k$ (Schritt 17), so folgt daher

$$V^{k+1}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t_k,$$

d.h.

$$V^{\infty}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t_k,$$

und falls $t_{k+1} = 0$ (Schritt 18), so folgt

$$V^{k+1}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) = \{0\} = V^{\infty}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}).$$

Damit ist

$$V^{\infty}(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} r_1$$

gezeigt. Die zweite Behauptung in (ii) ergibt sich unmittelbar aus Lemma 1.

□

Bemerkung 1

Die Rechnungen in den Schritten 2, 8 und 14 lassen sich mit Hilfe von Householdertransformationen realisieren. Es sei r die Gesamtzahl der durchgeführten Householdertransformationen (r ist abhängig von den $s_k, \tilde{s}_k, t_k$). Dann folgt aus dem in Abschnitt 2.2. Gesagten, daß der Algorithmus MAIU sich *numerisch stabil* implementieren läßt, in dem Sinne, daß Matrizen $\hat{Q} \in O(n)$, $E_A \in \mathbb{R}^{n \times n}$, $E_B \in \mathbb{R}^{n \times m}$, $E_H \in \mathbb{R}^{\ell \times n}$ existieren mit

$$\begin{aligned}\hat{Q}^T (A + E_A) \hat{Q} &= \hat{A} \\ \hat{Q}^T (B + E_B) &= \hat{B} \\ (H + E_H) \hat{Q} &= \hat{H},\end{aligned}$$

wobei

$$\left\| \begin{bmatrix} E_A & E_B \\ E_H & O \end{bmatrix} \right\|_F \leq 2r(1+12.36\epsilon) 2^r 12.36 \cdot \epsilon \left\| \begin{bmatrix} A & B \\ H & O \end{bmatrix} \right\|_F.$$

Bemerkung 2

Weder die Matrizen $\hat{A}$, $\hat{B}$, $\hat{H}$, noch die Indizes r_1, r_2, r_3 sind durch die Struktur in Lemma 2 (iii) eindeutig bestimmt.

Bemerkung 3

Es wird sich zeigen, daß zusätzlich zu den berechneten $\hat{A}$, $\hat{B}$, $\hat{H}$, Q , r_1 , r_2 , r_3 oftmals auch Matrizen der Form $\hat{G} = Q^T G$ und $\hat{C} = CQ$ benötigt werden, wobei $G \in \mathbb{R}^{n \times q}$

und $C \in \mathbb{R}^{p \times n}$ gegeben sind. Der Algorithmus MAIU läßt sich leicht so modifizieren, daß diese Matrizen in den Schritten 3, 7, 10, und 16 sukzessive gebildet werden. In diesem Fall hat der Algorithmus MAIU als Eingabe

$$n, m, \ell, q, p, A, B, H, G, C$$

und als Ausgabe

$$r_1, r_2, r_3, \hat{A}, \hat{B}, \hat{H}, \hat{G}, \hat{C}, Q.$$

Die Anwendung von dem modifizierten Algorithmus MAIU wird im folgenden abgekürzt durch

$$\text{MAIU}: (n, m, \ell, q, p, A, B, H, G, C) \mapsto (r_1, r_2, r_3, \hat{A}, \hat{B}, \hat{H}, \hat{G}, \hat{C}, Q).$$

5.2.3 Analyse von Systemen mit Algorithmus MAIU

In diesem Abschnitt werden die Störungsentkopplungsprobleme aus Abschnitt 5.1 untersucht.

5.2.3.1 Störungsentkopplung durch Zustandsrückführung

Die Lösbarkeit dieses Problems läßt sich durch den folgenden Algorithmus überprüfen.

Algorithmus 7

$$1. \text{ MAIU}: (n, m, \ell, q, -, A, B, H, G, -) \mapsto (r_1, r_2, r_3, \hat{A}, \hat{B}, \hat{H}, \hat{G}, -, -)$$

$$2. \text{ Partitioniere } \hat{G} = \begin{bmatrix} \hat{G}_1 \\ -\hat{G}_2 \end{bmatrix} \} r_1$$

3. SEPZ ist lösbar genau dann, wenn $\hat{G}_2 = 0$.

Beweis der Behauptung in Schritt 3

SEPZ ist lösbar genau dann, wenn $\text{im } G \subset V^\infty(A, \text{im } B, \ker H)$,
d.h.

$$\text{im}(Q^T G) \subset V^\infty(Q^T A Q, \text{im}(Q^T B), \ker(HQ)) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} r_1.$$

□

Bemerkung 4

In der vorliegenden Form hat Algorithmus 7 den Nachteil, daß der gesamte Algorithmus MAIU durchzuführen ist, um die Lösbarkeit zu entscheiden. Er kann jedoch leicht so modifiziert werden, daß die Lösbarkeit jeweils während der Durchführung des Algorithmus (nach Schritt 7) überprüft wird, und gestoppt wird, falls sich das Problem als unlösbar herausstellt.

5.2.3.2 Störungsentkopplung durch Ausgangsrückführung

Es werden zwei Algorithmen vorgestellt. Der erste Algorithmus ist offensichtlicher und einfacher zu beschreiben, während der zweite Algorithmus numerische und konzeptionelle Vorteile bietet.

Algorithmus 8

1. MAIU: $(n, p, q, -, -, A^T, C^T, G^T, -, -) \mapsto (r_1, r_2, r_3, -, -, -, -, Q)$

2. Partitioniere $Q = \underbrace{[Q_1]}_{r_1}, Q_2]$

3. MAIU: $(n, m, \ell, r_2 + r_3, -, A, B, H, Q_2, -) \mapsto (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3, -, -, -, \tilde{Q}_2, -, -)$

4. Partitioniere $\tilde{Q}_2 = \underbrace{\begin{bmatrix} \tilde{Q}_{21} \\ \tilde{Q}_{22} \end{bmatrix}}_{r_1 + r_2} \} \tilde{r}_1$

5. SEPA ist lösbar genau dann, wenn $\tilde{Q}_{22} = 0$.

Beweis der Behauptung in Schritt 5

Es gilt

$$\begin{aligned} W^\infty(A, \text{im } G, \ker C) &= [V^\infty(A^T, \text{im } C^T, \ker G^T)]^\perp \\ &= [QV^\infty(Q^T A^T Q, \text{im}(Q^T C^T), \ker(G^T Q))]^\perp \\ &= (Q \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \} r_1)^\perp = [\text{im } Q_1]^\perp = \text{im } Q_2, \end{aligned}$$

und daher ist SEPA genau dann lösbar, wenn

$$\text{im } Q_2 \subset V^\infty(A, \text{im } B, \ker H).$$

Die Behauptung folgt nun genau wie bei Algorithmus 7.

□

Der im folgenden beschriebenen alternative Algorithmus hat den Vorteil, daß die zweite Anwendung des Algorithmus MAIU an einem reduzierten Modell erfolgen kann, und daß er auch zur Synthese von Systemen herangezogen werden kann.

Algorithmus 9

1. MAIU: $(n, p, q, m, \ell, A^T, C^T, G^T, B, H) \mapsto (r_1, r_2, r_3, \tilde{A}^T, \tilde{C}^T, \tilde{G}^T, \tilde{B}, \tilde{H}, Q)$

2. Kommentar: Es gilt

$$\tilde{A} = \begin{bmatrix} * & * & 0 \\ * & * & * \\ * & * & * \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} * \\ * \\ * \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} 0 \\ * \\ * \end{bmatrix} \begin{matrix} r_1 \\ r_2 \\ r_3 \end{matrix},$$

$$\tilde{C} = [* \quad C' \quad 0],$$

$$\tilde{H} = \begin{bmatrix} * & * & * \\ r_1 & r_2 & r_3 \end{bmatrix},$$

wobei C' vollen Spaltenrang besitzt.

3. Vertausche die Blöcke in 2. mit einer Permutation P :

$$\hat{A} = P^T \tilde{A} P = \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & * & A' \end{bmatrix}, \quad \hat{B} = P^T \tilde{B} = \begin{bmatrix} * \\ * \\ B' \end{bmatrix}, \quad \hat{C} = P^T \tilde{C} = \begin{bmatrix} * \\ * \\ 0 \end{bmatrix} \begin{matrix} r_3 \\ r_2 \\ r_1 \end{matrix},$$

$$\hat{C} = \tilde{C} P = [0 \quad C' \quad *],$$

$$\hat{H} = H P = \begin{bmatrix} * & * & H' \\ r_3 & r_2 & r_1 \end{bmatrix}.$$

4. MAIU: $(r_1, m, \ell, -, -, A', B', H', -, -) \mapsto (r_1', r_2', r_3', A'', B'', H'', -, -, Q')$

5. Bilde $\hat{Q} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & Q' \end{bmatrix} \begin{matrix} r_3 \\ r_2 \\ r_1 \end{matrix}.$

6. Berechne $\bar{A} = \hat{Q}^T \hat{A} \hat{Q}$, $\bar{B} = \hat{Q}^T \hat{B}$, $\bar{C} = \hat{Q}^T \hat{C}$, $\bar{C} = \hat{C} \hat{Q}$, $\bar{H} = \hat{A} \hat{Q}$.

7. Kommentar: Es gilt

$$\bar{A} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} & \bar{A}_{14} & \bar{A}_{15} \\ \bar{A}_{21} & \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} & \bar{A}_{25} \\ 0 & \bar{A}_{32} & \bar{A}_{33} & \bar{A}_{34} & \bar{A}_{35} \\ 0 & \bar{A}_{42} & \bar{A}_{43} & \bar{A}_{44} & \bar{A}_{45} \\ 0 & \bar{A}_{52} & 0 & \bar{A}_{54} & \bar{A}_{55} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \\ \bar{B}_3 \\ \bar{B}_4 \\ 0 \end{bmatrix}, \quad \bar{G} = \begin{bmatrix} \bar{G}_1 \\ \bar{G}_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \} r_3 \\ \} r_2 \\ \} r'_1 \\ \} r'_2 \\ \} r'_3 \end{matrix},$$

$$\bar{C} = \begin{bmatrix} 0 & \bar{C}_2 & \bar{C}_3 & \bar{C}_4 & \bar{C}_5 \end{bmatrix},$$

$$\bar{H} = \begin{bmatrix} \bar{H}_1 & \bar{H}_2 & 0 & \bar{H}_4 & \bar{H}_5 \\ \underbrace{\phantom{\bar{H}_1}}_{r_3} & \underbrace{\phantom{\bar{H}_2}}_{r_2} & \underbrace{}_{r'_1} & \underbrace{\phantom{\bar{H}_4}}_{r'_2} & \underbrace{\phantom{\bar{H}_5}}_{r'_3} \end{bmatrix},$$

wobei $\bar{C}_2 = C'$ vollen Spaltenrang und $\bar{B}_4$ vollen Zeilenrang besitzt.

EPA ist lösbar genau dann, wenn $\bar{A}_{52} = 0$, $\bar{H}_1 = 0$ und $\bar{H}_2 = 0$.

Beweis der Behauptung in Schritt 8 werden die folgenden Überlegungen benutzt.

Bemerkung 5

Es $W^\infty(A, \text{im } G, \ker C) \subset V^\infty(A, \text{im } B, \ker H)$, so gilt in Schritt 2

Algorithmus 9:

$$\tilde{H} = \begin{bmatrix} * & 0 & 0 \\ \underbrace{}_{r_1} & \underbrace{}_{r_2} & \underbrace{}_{r_3} \end{bmatrix}.$$

Dieses folgt sofort aus

$$\begin{aligned} \text{im} \begin{bmatrix} 0 \\ I \end{bmatrix} \}_{r_2+r_3} &= W^\infty(\tilde{A}, \text{im } \tilde{G}, \ker \tilde{C}) \\ &\subset V^\infty(\tilde{A}, \text{im } \tilde{B}, \ker \tilde{H}) \subset \ker \tilde{H}. \end{aligned}$$

Bemerkung 6

Falls die in Q' zusammengefaßten Transformationen des Algorithmus MAIU in Schritt 4 von Algorithmus 9 jeweils während der Durchführung von MAIU auf $\bar{A}$, $\bar{B}$, $\bar{G}$, $\bar{C}$, $\bar{H}$ angewendet werden, so ergibt sich in Schritt 7 des Algorithmus MAIU die folgende Struktur:

$$\begin{aligned} \bar{A}^k &= \begin{array}{ccccc} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & A_{11}^k & A_{12}^k & A_{13}^k \\ 0 & * & A_{21}^k & A_{22}^k & A_{23}^k \\ 0 & z^k & 0 & A_{32}^k & A_{33}^k \end{array} \begin{array}{l} \left. \begin{array}{c} \\ \\ \end{array} \right\} r_3 \\ \left. \begin{array}{c} \\ \\ \end{array} \right\} r_2 \\ \left. \begin{array}{c} \\ \\ \end{array} \right\} t_k \\ \left. \begin{array}{c} \\ \\ \end{array} \right\} s_k \end{array} \quad \begin{array}{c} \bar{B}^k = \begin{array}{c} * \\ * \\ B_1^k \\ B_2^k \\ 0 \end{array} \quad \bar{G}^k = \begin{array}{c} * \\ * \\ 0 \\ 0 \\ 0 \end{array} \end{array} \\ \bar{C}^k = \begin{array}{ccccc} 0 & * & * & * & * \end{array} \\ \bar{H}^k = \begin{array}{ccccc} * & * & 0 & H_2^k & H_3^k \end{array} \end{aligned}$$

Wäre jetzt $z^{k+1} \neq 0$, so gäbe es ein $x = \begin{bmatrix} 0 \\ x_2 \\ 0 \end{bmatrix} \begin{matrix} \} r_3 \\ \} r_2 \end{matrix} \in \mathbb{R}^n$

mit $z_2 = z^{k+1} x_2 \neq 0$. Für dieses x hätte man dann einerseits

$$x \in W^\infty(\bar{A}^{k+1}, \text{im } \bar{G}^{k+1}, \ker \bar{C}^{k+1}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{matrix} \} r_3 + r_2 \\ \} \end{matrix}$$

und andererseits

$$\begin{aligned} \bar{A}^{k+1} x &= \begin{bmatrix} * \\ z_2 \end{bmatrix} \notin \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{matrix} \} r_3 + r_2 + t_k \\ \} \end{matrix} + \text{im } \bar{B}^{k+1} \\ &= V^k(\bar{A}^{k+1}, \text{im } \bar{B}^{k+1}, \ker \bar{H}^{k+1}) + \text{im } \bar{B}^{k+1}, \end{aligned}$$

d.h.

$$x \notin V^{k+1}(\bar{A}^{k+1}, \text{im } \bar{B}^{k+1}, \ker \bar{H}^{k+1}) \supset V^\infty(\bar{A}^{k+1}, \text{im } \bar{B}^{k+1}, \ker \bar{H}^{k+1}).$$

Dieses steht aber im Widerspruch zur Annahme des Lemmas, also gilt $z^{k+1} = 0$. Die Behauptung (ii) folgt jetzt genauso wie (i) in Proposition 2. □

Beweis der Behauptung in Schritt 8 von Algorithmus 9

Es sei SEPA lösbar, d.h.

$$W^\infty(A, \text{im } G, \ker C) \subset V^\infty(A, \text{im } B, \ker H).$$

Bemerkung 5 und Lemma 3 implizieren dann $\bar{H}_1 = 0$, $\bar{H}_2 = 0$ und $\bar{A}_{52} = 0$. Andererseits, falls $\bar{H}_1 = 0$, $\bar{H}_2 = 0$ und

$\bar{A}_{52} = 0$, so ist $\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{matrix} \} r_3 + r_2 \\ \} \end{matrix}$ ein $(\bar{C}, \bar{A})$ -invarianter

Unterraum, der im $\bar{G}$ enthält, und $\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{matrix} \} r_3 + r_2 + r_1' \\ \} \end{matrix}$ ist

ein $(\bar{A}, \bar{B})$ -invarianter Unterraum in $\ker \bar{H}$. □

5.2.3.3 Störungsentkopplung und Polzuweisung durch Zustandsrückführung

Die Lösbarkeit von SEPPZ läßt sich wie folgt überprüfen:

Algorithmus 10

1. MAIU: $(n, m, \ell, -, -, A, B, H, -, -) \rightarrow (r_1, r_2, r_3, -, -, -, -, Q)$.
2. Partitioniere $Q = [\underbrace{Q_1}_{\bar{r}_1}, Q_2]$.
3. MAIU: $(n, r_2 + r_3, m, q, -, A^T, Q_2, B^T, G, -) \rightarrow (\bar{r}_1, \bar{r}_2, \bar{r}_3, \bar{A}^T, \bar{Q}_2, \bar{B}^T, \bar{G}, -, \bar{Q})$
4. Partitioniere $\bar{Q}_2^T = [\underbrace{*}_{\bar{r}_1} z_1]$, und schiebe die Spalten
von z_1 nach links:

$$\bar{Q}_2^T \hat{Q} = [\underbrace{*}_{\bar{r}_1} \underbrace{z_2}_{:t} \quad 0],$$

wobei z_2 vollen Spaltenrang hat und $\hat{Q} = \begin{bmatrix} I & 0 \\ 0 & * \end{bmatrix} \}_{\bar{r}_1} \in O(n)$.

5. Bilde $\hat{G} = \hat{Q}^T \bar{G}$ und partitioniere $\hat{G} = \begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \end{bmatrix} \}_{\bar{r}_1 + t}$.

6. SEPPZ ist lösbar genau dann, wenn $\hat{G}_2 = 0$.

Beweis der Behauptung in Schritt 6

Es gilt $V^\infty(A, \text{im } B, \ker H) = \text{im } Q_1 = \ker Q_2^T$, und daher ist nach Abschnitt 5.1.3 SEPPZ genau dann lösbar, wenn

$$\begin{aligned}
 \text{im } G &\subset R^\infty(A, \text{im } B, V^\infty(A, \text{im } B, \ker H)) \\
 &= R^\infty(A, \text{im } B, \ker Q_2^T) \\
 &= [V^\infty(A^T, \text{im } Q_2, \ker B^T)]^\perp \cap \ker Q_2^T \\
 &= [\bar{Q}V^\infty(\bar{A}^T, \text{im } \bar{Q}_2, \ker \bar{B}^T)]^\perp \cap \ker Q_2^T \\
 &= \bar{Q}(\text{im} \begin{bmatrix} 0 \\ I \end{bmatrix} \bar{r}_1 \cap \ker \bar{Q}_2^T) \\
 &= \bar{Q} \hat{Q}(\text{im} \begin{bmatrix} 0 \\ I \end{bmatrix} \bar{r}_1 \cap \ker[* \ Z_2 \ 0]) \\
 &= \bar{Q} \hat{Q}(\text{im} \begin{bmatrix} 0 \\ I \end{bmatrix} \bar{r}_1^{+t}).
 \end{aligned}$$

□

Algorithmus 10 ist numerisch stabil in dem Sinne, daß alle berechneten Matrizen orthogonal ähnlich (bzw. äquivalent) zu leicht gestörten Ausgangsdaten sind.

5.2.3.4 Fast-Störungsentkopplung mit strenger Stabilisierung durch Zustandsrückführung

Mit Hilfe des folgenden Algorithmus läßt sich die Lösbarkeit von FSEPSSZ überprüfen.

Algorithmus 11

1. MAIU: $(n, \ell, m, q, -, A^T, H^T, B^T, G, -) \mapsto (r_1, r_2, r_3, -, -, \hat{G}, -)$

2. Partitioniere $\hat{G} = \begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \end{bmatrix} \bar{r}_1$

3. FSEPSSZ ist lösbar genau dann, wenn $\hat{G}_1 = 0$

Beweis der Behauptung in Schritt 3

Es bezeichne Q die orthogonale Transformationsmatrix aus Schritt 1. Dann ist FSEPSSZ lösbar genau dann, wenn

$$\begin{aligned}
 \text{im } \hat{G} &= \text{im } Q^T G \subset Q^T W^\infty(A, \text{im } B, \ker H) \\
 &= Q^T [V^\infty(A^T, \text{im } H^T, \ker B^T)]^\perp \\
 &= [Q^T V^\infty(A^T, \text{im } H^T, \ker B^T)]^\perp \\
 &= [V^\infty(Q^T A^T Q, \text{im}(Q^T H^T), \ker(B^T Q))]^\perp \\
 &= [\text{im} \begin{bmatrix} I \\ 0 \end{bmatrix}]^\perp r_1 \\
 &= \text{im} \begin{bmatrix} 0 \\ I \end{bmatrix} r_1.
 \end{aligned}$$

□

5.3. Synthese von Systemen mit Hilfe von Matrix-Algorithmen

In diesem Abschnitt werden Algorithmen zur Lösung der Störungsentkopplungsprobleme aus Abschnitt 5.1. vorgestellt. Diese Algorithmen bauen teilweise auf denen von Abschnitt 5.2. auf, teilweise sind aber auch neue Algorithmen nötig. Dieses liegt daran, daß manche der in Abschnitt 5.2. berechneten Zustandsdarstellungsformen nicht genügend "Struktur" besitzen, um darauf einen Entwurfsalgorithmus aufzubauen. Als Illustration betrachte man die Algorithmen 7 und 11, die ein gleiches Maß an Struktur liefern. In diesem Abschnitt wird sich zeigen, daß diese Struktur genügt, um SEPZ zu lösen, aber nicht genügt, um das wesentlich komplexere Problem FSEPSSZ zu lösen.

Falls die Struktur der Zustandsdarstellungsformen aus Abschnitt 5.2 nicht ausreicht, werden in diesem Abschnitt Zustandsdarstellungsformen mit mehr Struktur entwickelt. Diese zusätzliche Struktur wird allerdings teilweise durch nicht-orthogonale Transformation "erkauft", so daß die numerische Stabilität einiger Algorithmen aus diesem Abschnitt nicht mehr sichergestellt ist. Um die Rechenfehler so gering wie möglich zu halten, werden jedoch vorzugsweise orthogonale Transformationen benutzt. Auch wird mit den Matrizen $A, B, \dots$ selbst gearbeitet und nicht etwa mit der Erreichbarkeitsmatrix etc.

5.3.1. Störungsentkopplung durch Zustandsrückführung

In diesem Abschnitt wird das in (Aplevich 1979) und (van Dooren 1981) vorgeschlagene Syntheseverfahren wiedergegeben. Der Algorithmus MAIU liefert Matrizen $\hat{A}, \hat{B}, \hat{A}$ mit der Struktur aus Lemma 2. Wird zusätzlich G transformiert, so erhält man

$$\hat{G} = Q^T G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}. \quad (11)$$

Ist SEPZ lösbar, was jetzt angenommen wird, so gilt zusätzlich $G_2 = 0$ und $G_3 = 0$. Die Struktur von $\hat{A}, \hat{B}, \hat{A}, \hat{G}$ läßt sich jetzt aber sehr einfach zur Lösung von SEPZ ausnutzen: Wählt man $F_1 \in \mathbb{R}^{m \times r_1}$ so, daß die linearen Gleichungssysteme

$$B_2 F_1 = - A_{21} \quad (12)$$

erfüllt sind (B_2 hat vollen Zeilenrang!), dann ist

$$F = \underbrace{[F_1]}_{r_1} \underbrace{[0]}_{r_2} \underbrace{[0]}_{r_3} Q^T \quad (13)$$

eine Lösung von SEPZ. Dieses Vorgehen ist die in der Einleitung beschriebene *Zweischrittmethode*: Zunächst wird das Problem in ein einfacheres transformiert, das dann gelöst wird.

Da alle Teilschritte des Algorithmus numerisch stabil sind, kann man den Algorithmus als zuverlässig bezeichnen. Ohne weitere Einschränkungen an das Problem kann man jedoch keine numerische Stabilität in dem restriktiven Sinne von Abschnitt 2.2. zeigen. Man beachte auch, daß es bislang noch keine Definition eines gut konditionierten Störungsentkopplungsproblems gibt.

5.3.2. Störungsentkopplung durch Ausgangsrückführung

Das Problem SEPA sei lösbar. Dann läßt sich das System mit Hilfe des Algorithmus 9 in die in Schritt 7 beschriebene Form transformieren, wobei $\bar{A}_{52} = 0$, $\bar{H}_1 = 0$ und $\bar{H}_2 = 0$. Diese Zustandsdarstellungsform soll nun zur Berechnung des dynamischen Reglers herangezogen werden. Zunächst wird jedoch die wesentliche Idee kurz für den Spezialfall $\bar{B}_1, \bar{B}_2, \bar{B}_3, \bar{C}_3, \bar{C}_4, \bar{C}_5 = 0$ erläutert. Dann besitzt das Signalflußdiagramm des Systems die Struktur aus Bild 1.

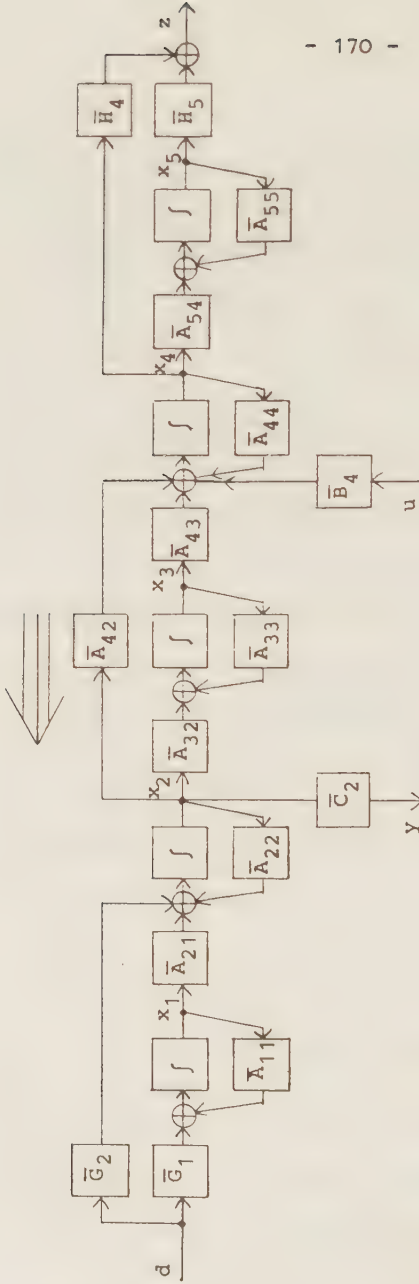


Bild 1 (der waagerechte Pfeil bedeutet, daß alle Kopplungen von x_1 nach $\dot{x}_j$ ($j < i$) vorhanden sind)

Man erkennt, daß eine Störung, um von $\bar{d}$ nach z zu gelangen, das Teilsystem zwischen x_2 und $\dot{x}_4$ vollständig durchlaufen muß. Die Strategie wird jetzt sein, dieses Teilsystem in einem dynamischen Pegler von y nach u (mit negativem Vorzeichen) zu duplizieren. Jedes von x_2 ausgehende Signal wird sich dann bei $\dot{x}_4$ auslöschen.

Zur Konstruktion des Reglers wähle zunächst N so, daß $\bar{B}_4 N \bar{C}_2 = -\bar{A}_{42}$. Das geschlossene System (bei unbestimmten M, E, F) läßt sich dann in die folgende Form permutieren:

$$\tilde{A} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} & B_3 M & * & * \\ 0 & E \bar{C}_2 & E \bar{C}_3 & F & * & * \\ 0 & 0 & \tilde{A}_{53} & \bar{B}_4 M & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \quad \tilde{G} = \begin{bmatrix} * \\ * \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

$$\tilde{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

Wähle jetzt E, M und F so, daß

$$\begin{aligned} E \bar{C}_2 &= \tilde{A}_{32} \\ \bar{B}_4 M &= -\tilde{A}_{53} \\ F &= \tilde{A}_{33} + B_3 M - E \bar{C}_3. \end{aligned}$$

Addiert man jetzt die vierte Block-Spalte zur dritten und subtrahiert die dritte Block-Spalte von der vierten, so erhält man ein zu (14) ähnliches System der Struktur

$$A' = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}, \quad G' = \begin{bmatrix} * \\ * \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (15)$$

$$H' = [0 \ 0 \ 0 \ 0 \ * \ *].$$

Die Transferfunktion des Systems in (15) ist jedoch identisch Null.

Bei der obigen Konstruktionsmethode sind neben den orthogonalen Transformationen aus Algorithmus 9 nur lineare Gleichungssysteme zu lösen. Daher kann man die Methode - obwohl nicht numerisch stabil - als zuverlässig bezeichnen.

5.3.3. Störungsentkopplung mit Polzuweisung durch Zustandsrückführung

Es ist nicht klar, wie die von Algorithmus 10 gelieferte Zustandsdarstellungsform zur Berechnung einer Rückführung herangezogen werden kann. Satz 4 aus Abschnitt 5.1.3. liefert jedoch eine mögliche Methode zur Berechnung von F , die in (Aplevich 1979) und (van Dooren 1981) vorgeschlagen und hier kurz wiederholt wird.

Es wird zunächst ein störungsentkoppelnder Regler wie in Abschnitt 5.3.1. berechnet. Man erhält dann

$$Q^T(A+BF)Q = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix}, \quad Q^TB = \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \\ 0 \end{bmatrix}, \quad Q^TG = \begin{bmatrix} \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_3 \end{bmatrix}, \quad (16)$$

$$HQ = [0 \quad \tilde{H}_2 \quad \tilde{H}_3].$$

Im nächsten Schritt (Berechnung von $V_{\ker H}^* \cap \text{im } B$) wird eine orthogonale Eingangstransformation $R \in O(m)$ so bestimmt, daß

$$Q^TBR = \begin{bmatrix} \tilde{B}_{11} & \tilde{B}_{12} \\ 0 & \tilde{B}_{22} \\ \underbrace{0}_{m-r_2} & \underbrace{0}_{r_2} \end{bmatrix} \begin{matrix} \} r_1 \\ \} r_2 \\ \} r_3 \end{matrix}, \quad (17)$$

wobei $\tilde{B}_{22} \in Gl(r_2)$. Den Raum $R_{\ker H}^*$ erhält man dann als den erreichbaren Teil von $(\tilde{A}_{11}, \tilde{B}_{11})$, den man beispielsweise mit einem der in Abschnitt 3.2.1. beschriebenen Hessenberg-Algorithmen bestimmen kann. Man erhält dann das folgende System:

$$\bar{Q}^T(A+BF)Q = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} & \bar{A}_{14} \\ 0 & \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & 0 & \tilde{A}_{22} & \tilde{A}_{23} \\ 0 & 0 & \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix}, \quad \bar{Q}^T B R = \begin{bmatrix} \bar{B}_{11} & \bar{B}_{12} \\ 0 & \bar{B}_{22} \\ 0 & \tilde{B}_{22} \\ 0 & 0 \end{bmatrix}, \quad \bar{Q}^T G = \begin{bmatrix} \bar{G}_1 \\ \bar{G}_2 \\ \tilde{G}_2 \\ \tilde{G}_3 \end{bmatrix},$$

$$H\bar{Q} = [0 \quad 0 \quad \tilde{H}_2 \quad \tilde{H}_3],$$

wobei $(\bar{A}_{11}, \bar{B}_{11})$ erreichbar ist, $R_{\ker H}^* = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} t$,
und t die Dimension von $\bar{A}_{11}$ bezeichnet.

SEPPZ ist also lösbar genau dann, wenn $\bar{G}_2 = 0$, $\tilde{G}_2 = 0$,
 $\tilde{G}_3 = 0$ und (A, B) erreichbar ist. In diesem Fall ist das
System bereits störungsentkoppelt, und ein zusätzlicher
polzuweisender Regler läßt sich einfach entwerfen, indem
die ersten $m - r_2$ Eingänge (in der neuen Basis) zur Pol-
zuweisung in $\bar{A}_{11}$ und die letzten r_2 Eingänge zur Polzu-
weisung in

$$\begin{bmatrix} \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix} \quad (19)$$

herangezogen werden.

Wie bereits von *van Dooren* (1981) bemerkt, ist dieses Vor-
gehen aus numerischer Sicht insofern heikel, als daß die
Rückführungsoperationen nicht am Ende des Algorithmus aus-
geführt werden. Eine Rückführungsoperation

$$[A, B] \mapsto [A+BF, B] = [A, B] \begin{bmatrix} I & 0 \\ F & I \end{bmatrix} \quad (20)$$

läßt sich nicht numerisch stabil ausführen, und daher läßt sich auch (18) nicht numerisch stabil berechnen. Ferner, falls die Matrix $\tilde{B}_2$ in (16) schlecht konditioniert (bezüglich der Lösung von linearen Gleichungssystemen) ist, kann die Form (18) beliebig schlecht konditioniert sein.

Eine schlecht konditionierte Matrix $\tilde{B}_2$ bedeutet jedoch, daß es schwierig ist, den Raum $V_{\ker H}^*$ A-invariant zu machen. Insofern sind die aufgrund einer schlecht konditionierten Matrix $\tilde{B}_2$ auftretenden Probleme nicht so sehr dem Algorithmus, als vielmehr der Problemstellung zuzuordnen. Daher kann man bei einem "gut konditionierten" SEPPZ davon ausgehen, daß $\tilde{B}_2$ ebenfalls gut konditioniert ist. In diesem Fall kann man zusätzlich erwarten, daß die Matrix F nicht zu groß ist, so daß die Operation (20) stabil ist. Insgesamt kann man den Algorithmus dann als zuverlässig bezeichnen. Man beachte jedoch, daß trotz gut konditioniertem $\tilde{B}_2$ die Form (18) beliebig schlecht konditioniert sein kann: Die in Kapitel 3 beschriebenen Hessenberg-Formen sind Spezialfälle dieser Form !

Bei den obigen Überlegungen wurde suggeriert, daß, falls $V_{\ker H}^*$ "schwierig" A-invariant zu machen ist, SEPZ schlecht konditioniert ist. Dieses ist jedoch nicht der Fall, da zur Lösung von SEPZ lediglich irgendein (A,B) -invarianter Unterraum U mit $\text{im } G \subset U \subset \ker H$ A-invariant zu machen ist. Es ist ein interessantes Problem, einen "gut konditionierten"

Unterraum U mit den obigen Eigenschaften zu finden. Eventuell können dabei die Ergebnisse in (Hinrichsen, Münzner und Prätzel-Wolters 1981) von Nutzen sein.

In (van Dooren 1981) wird eine zu dem obigen Vorgehen alternative Methode vorgeschlagen, bei dem die Rückführungsoperation an das Ende des Algorithmus verschoben wird. Dieses geschieht derart, daß ausgehend von der Form in Lemma 2 nicht (16), sondern direkt (17) und (18) berechnet werden, wobei jeweils antizipiert wird, daß später die störungsentkoppelnde Rückführung noch angewendet wird. Für Details vergleiche (van Dooren 1981). Da dieser Algorithmus jedoch sehr viel aufwendiger ist, ist zweifelhaft, ob dadurch bessere Resultate erzielt werden können, zumal die Problematik einer schlecht konditionierten Matrix $\tilde{B}_2$ dabei nicht umgegangen wird.

Zum Abschluß dieses Abschnittes wird eine weitere Methode zur Lösung von SEPPZ vorgestellt, die sich von der bereits beschriebenen insofern unterscheidet, als daß anstatt der zu (16) führenden Rückführungsoperation eine nicht-orthogonale Ähnlichkeitstransformation durchgeführt wird. Der Algorithmus hat die gleichen numerischen Eigenschaften wie der zu den Gleichungen (16) bis (18) führende, liefert aber eine Zustandsdarstellungsform, die in Abschnitt 5.4. zur Charakterisierung der Lösbarkeit von SEPPZ benötigt wird.

Ausgegangen wird wieder von der in Lemma 2 beschriebenen Zustandsdarstellungsform. Durch eine orthogonale Eingangstransformation $R \in O(n)$ wird Q^TB dann in die Form (17) gebracht, wobei $\tilde{B}_{22} \in Gl(r_2)$. Sodann wird $K \in R^{r_1 \times r_2}$ so bestimmt, daß $K\tilde{B}_{22} = -\tilde{B}_{12}$. Mit

$$\bar{S} = \begin{bmatrix} I & -K & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \quad (21)$$

hat das System dann die folgende Struktur:

$$\bar{S}^{-1}Q^T A Q \bar{S} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ \tilde{A}_{21} & \tilde{A}_{22} & \tilde{A}_{23} \\ 0 & \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix}, \quad \bar{S}^{-1}Q^T B R = \begin{bmatrix} \tilde{B}_{11} & 0 \\ 0 & \tilde{B}_{22} \\ 0 & 0 \end{bmatrix}, \quad \bar{S}^{-1}Q^T G = \begin{bmatrix} \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_3 \end{bmatrix}, \quad (22)$$

$$H Q \bar{S} = [0 \quad \tilde{H}_2 \quad \tilde{H}_3].$$

Diese Zustandsdarstellungsform hat den Vorteil, daß die zur Störungsentkopplung und zur Polzuseisung benötigten Eingänge entkoppelt sind. Als nächster Schritt wird durch eine orthogonale Transformation $P \in O(n)$ der erreichbare Teil von $(\tilde{A}_{11}, \tilde{B}_{11})$ bestimmt:

$$S^{-1} A S = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} & \bar{A}_{13} & \bar{A}_{14} \\ 0 & \bar{A}_{22} & \bar{A}_{23} & \bar{A}_{24} \\ \bar{A}_{31} & \bar{A}_{32} & \bar{A}_{33} & \bar{A}_{34} \\ 0 & 0 & \bar{A}_{43} & \bar{A}_{44} \end{bmatrix}, \quad S^{-1} B R = \begin{bmatrix} \bar{B}_{11} & 0 \\ 0 & 0 \\ 0 & \bar{B}_{32} \\ 0 & 0 \end{bmatrix}, \quad S^{-1} G = \begin{bmatrix} \bar{G}_1 \\ \bar{G}_2 \\ \bar{G}_3 \\ \bar{G}_4 \end{bmatrix}, \quad (23)$$

$$H S = [0 \quad 0 \quad \bar{H}_3 \quad \bar{H}_4],$$

wobei $(\bar{A}_{11}, \bar{B}_{11})$ erreichbar ist und $S = Q\bar{S}P$. Wegen Satz 4 ist SEPPZ lösbar genau dann, wenn (A, B) erreichbar ist und $\bar{G}_2 = 0$, $\bar{G}_3 = 0$, $\bar{G}_4 = 0$. In diesem Fall erhält man den gewünschten Zustandsregler, indem man die ersten Eingänge zur Polzuweisung in $\bar{A}_{11}$ nutzt, und mit den letzten Eingängen $\bar{A}_{31}$ auslöscht und die Pole des restlichen Systems zuweist.

5.3.4. Fast-Störungsentkopplung

Der vorherige Abschnitt hat gezeigt, daß die durch orthogonale Transformationen erhaltenen Zustandsdarstellungsformen, an denen sich die Lösbarkeit des Problems entscheiden läßt, für Synthesefragen teilweise zu wenig Struktur besitzen. Mit anderen Worten: Eine (orthogonale) Basis charakteristischer Unterräume (z.B. $R_{\ker H}^*$) ist i.a. nicht ausreichend, um bestimmte Syntheseprobleme (z.B. SEPPZ) konkret lösen zu können.

Diese Problematik tritt auch bei FSEPSSZ auf. Wie in Abschnitt 5.2.3.4 dargestellt, kann man sich mit dem Algorithmus MAIU relativ einfach eine Basis von $R_{b, \ker H}^*$ verschaffen und damit die Lösbarkeit von FSEPSSZ entscheiden. Die daraus resultierende Darstellung

$$\begin{aligned} \hat{A} &= \begin{bmatrix} \overbrace{A_{11}}^{r_1} & \overbrace{A_{12}}^{r_2} & \overbrace{0}^{r_3} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ B_2 \\ B_3 \end{bmatrix}, \quad \hat{G} = \begin{bmatrix} 0 \\ G_2 \\ G_3 \end{bmatrix} \begin{matrix} \} r_1 \\ \} r_2 \\ \} r_3 \end{matrix} \\ \hat{H} &= [H_1 \quad H_2 \quad 0] \end{aligned} \quad (24)$$

mit

$$\operatorname{rg} H_2 = r_2, \quad (25)$$

$$\operatorname{rg} \begin{bmatrix} A_{23} & B_2 \\ A_{33}-sI & B_3 \end{bmatrix} = r_2+r_3 \quad \forall s \in \mathbb{C} \quad (26)$$

und

$$R_{b, \ker H}^* = \operatorname{im} \begin{bmatrix} 0 \\ I \end{bmatrix} \} r_2+r_3 \quad (27)$$

kann aber noch nicht auf direkte Weise zur Berechnung der Rückführung herangezogen werden.

Es wird jetzt ein zu MAIU alternativer Algorithmus vorgeschlagen, der auch nicht-orthogonale Transformationen benutzt, dafür aber, anstatt der relativ abstrakten Bedingung (26), eine Darstellung liefert, bei der (26) aus der Struktur der Matrizen offensichtlich ist. Diese Struktur wird dann zur Berechnung der Rückführung ausgenutzt.

Um den Unterschied des zu beschreibenden Algorithmus zum Algorithmus MAIU zu verdeutlichen, betrachte man zunächst noch einmal den Algorithmus MAIU.

Zunächst wird die Matrix H in die Form

$$\begin{bmatrix} 0 & \tilde{H}^0 \\ t_1 & \end{bmatrix}$$

transformiert, wobei $\tilde{H}^0$ vollen Spaltenrang besitzt. Im Laufe des Algorithmus erhält die Matrix H jedoch die Form

$$\left[\underbrace{0}_{t_k} \tilde{H}^k \right],$$

und, da $t_1 > t_2 > t_3 > \dots$, die zunächst sehr einfache Struktur $[0 \tilde{H}^0]$ wird langsam wieder zerstört. Insbesondere haben die $\tilde{H}^k$ für $k \geq 1$ natürlich nicht mehr vollen Spaltenrang. Das Ziel des folgenden Algorithmus MASIU (Matrix-Algorithmus mit Struktur zur Berechnung von in-varianten Unterräumen) ist es, diesen Effekt zu vermeiden, d.h. einmal hergestellte Strukturen sollen nicht mehr zerstört werden.

Algorithmus MASIU

1. Eingabe $n, m, \ell \geq 1$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $H \in \mathbb{R}^{\ell \times n}$.
2. Schiebe die Spalten von H nach rechts mit $S^1 \in O(n)$:

$$HS^1 = \left[0 \underbrace{E_0}_{:t_1} \right],$$

wobei E_0 vollen Spaltenrang t_1 besitzt.

3. Berechne

$$A^1 = (S^1)^T A S^1$$

$$B^1 = (S^1)^T B$$

$$H^1 = HS^1.$$

4. $k := 1$, $R^1 := I \in O(m)$.

5. Falls $t_1 = 0$, d.h. falls $H = 0$, so setze

$$\hat{A} = A^1, \hat{B} = B^1, \hat{H} = H^1, S = S^1, R = I \text{ und stoppe.}$$

6. Kommentar: Es gilt

$$(i) \quad A^k = (S^k)^{-1} A S^k$$

$$B^k = (S^k)^{-1} B R^k$$

$$H^k = H S^k$$

$$S^k \in \text{Gl}(n)$$

$$R^k \in O(m)$$

$$(ii) \quad 0 < t_i \text{ für } i \in \underline{k}, \quad 0 < s_i \leq t_i \text{ für } i \in \underline{k-1}, \\ t_1 + t_2 + \dots + t_k \leq n$$

(iii) A^k, B^k, H^k haben die in Bild 2¹⁾ dargestellte Struktur, wobei $E_0, E_1, \dots, E_{k-1}$ vollen Spaltenrang besitzen und $D_1, D_2, \dots, D_{k-1}$ invertierbar sind.

7. Falls $k > 1$ und $(t_1 - s_1) + (t_2 - s_2) + \dots + (t_{k-1} - s_{k-1}) = m$, so setze $s_k := t_k$ und gehe zu 16.

8. Schiebe die Zeilen von X_k nach oben mit $\tilde{Q}_2 \in O(t_k)$:

$$\tilde{Q}_2^T X_k = \begin{bmatrix} \tilde{X}_k \\ 0 \end{bmatrix}_{:s_k},$$

wobei $\tilde{X}_k$ vollen Zeilenrang besitzt.

¹⁾ Bild 2 ist insofern mißverständlich, als dort suggeriert wird, daß $s_i \leq t_{i+1}$. In Wirklichkeit gilt aber $s_i \geq t_{i+1}$, da die Matrizen E_i vollen Spaltenrang besitzen. Die vorliegende Form wurde aus darstellerischen Gründen gewählt.

9. Bilde

$$\tilde{Q} = \begin{bmatrix} I & O & O \\ O & \tilde{Q}_2 & O \\ O & O & I \end{bmatrix} \}_{t_k} \in O(n).$$

$t_1+t_2+\dots+t_{k-1}$

10. Berechne

$$\tilde{A}^k = \tilde{Q}^T A^k \tilde{Q}$$

$$\tilde{B}^k = \tilde{Q}^T B^k$$

$$\tilde{H}^k = H^k \tilde{Q}$$

$$\tilde{S}^k = S^k \tilde{Q}.$$

11. Falls $s_k = t_k$, d.h. falls $x_k = 0$, so gehe zu 16.

12. Schiebe die Spalten von $\tilde{X}_k$ nach links mit

$$\tilde{R}_2 \in O(m - (t_1 - s_1) - (t_2 - s_2) - \dots - (t_{k-1} - s_{k-1})):$$

$$\tilde{X}_k \tilde{R}_2 = \begin{bmatrix} D_k \\ \underbrace{\hspace{2cm}}_{t_k - s_k} \end{bmatrix} O.$$

13. Bilde

$$\tilde{R} = \begin{bmatrix} I & O \\ O & \tilde{R}_2 \end{bmatrix} \in O(m).$$

14. Berechne

$$\tilde{B}^k = \tilde{B}^k \tilde{R}$$

$$R^{k+1} = R^k \tilde{R}.$$

15. Falls $s_k = 0$, d.h. falls x_k vollen Zeilenrang besitzt,

$$\text{so setze } \hat{A} = \tilde{A}^k, \hat{B} = \tilde{B}^k, \hat{H} = \tilde{H}^k, S = \tilde{S}^k, R = R^{k+1}$$

und stoppe.

16. Kommentar: Das System $\tilde{A}^k, \tilde{B}^k, \tilde{H}^k$ hat jetzt die gleiche Struktur wie in Bild 2, nur daß die Teilmatrix X_k ersetzt ist durch

$$\underbrace{\begin{bmatrix} D_k & \\ & O \end{bmatrix}}_{t_k - s_k} \Bigg\} s_k, \quad D_k \in Gl(t_k - s_k),$$

die Teilmatrix V_k (per Definition) ersetzt ist durch

$$\left\{ \begin{array}{c} \boxed{\quad * \quad} \\ \boxed{\quad W_k \quad} \end{array} \right\} s_k,$$

wobei W_k keine besondere Struktur besitzt, und die Teilmatrix Y_k (per Definition) ersetzt ist durch

$$t_k \left\{ \begin{array}{c} \boxed{\quad * \quad} \\ \boxed{\quad Z_k \quad} \end{array} \right\} s_k, \quad (28)$$

wobei Z_k keine besondere Struktur besitzt.

17. Wähle $T \in Gl(n)$ von der Form

$$T = \begin{bmatrix} I & O & O \\ O & I & * \\ O & O & I \end{bmatrix} \begin{array}{l} t_k \\ t_1 + t_2 + \dots + t_{k-1} \end{array},$$

so daß die Teilmatrix W_k in $T^{-1} \tilde{B}^k$ eliminiert wird.

18. Berechne

$$\bar{A}^k = T^{-1} \hat{A}^k T$$

$$\bar{B}^k = T^{-1} \hat{B}^k$$

$$\bar{H}^k = T^{-1} \hat{H}^k$$

$$\bar{S}^k = \hat{S}^k T \quad .$$

19. Kommentar: Das System besitzt noch die gleiche Struktur wie in Schritt 16; lediglich die Matrix Z_k hat sich verändert. Bezeichne die neue Matrix mit $\bar{Z}_k$.

20. Falls $t_1 + t_2 + \dots + t_k = n$, d.h. falls die Matrix in (28) nicht vorhanden ist, so setze $\hat{A} = \bar{A}^k$, $\hat{B} = \bar{B}^k$, $\hat{H} = \bar{H}^k$, $S = \bar{S}^k$, $R = R^{k+1}$ und stoppe.

21. Schiebe die Spalten von $\bar{Z}_k$ nach rechts mit $\bar{Q}_1 \in O(n - t_1 - t_2 - \dots - t_k)$:

$$\bar{Z}^k \bar{Q}_1 = \begin{bmatrix} 0 & \underbrace{E_k}_{:t_{k+1}} \end{bmatrix} ,$$

wobei E_k vollen Spaltenrang besitzt.

22. Bilde $\bar{Q} := \begin{bmatrix} \bar{Q}_1 & 0 \\ 0 & I \end{bmatrix} \in O(n)$.

23. Berechne

$$A^{k+1} := \bar{Q}^T \bar{A}^k \bar{Q}$$

$$B^{k+1} := \bar{Q}^T \bar{B}^k$$

$$H^{k+1} := \bar{H}^k \bar{Q}$$

$$S^{k+1} := \bar{S}^k \bar{Q} \quad .$$

24. Falls $t_{k+1} = 0$, d.h. falls $\bar{z}^k = 0$, so setze
 $\hat{A} = A^{k+1}$, $\hat{B} = B^{k+1}$, $\hat{H} = H^{k+1}$, $S = S^{k+1}$, $R = R^{k+1}$
 und stoppe.

25. $k \leftarrow k+1$

26. Gehe zu 6.

Proposition 3

Für das durch den Algorithmus berechnete System gilt

$$(i) \quad \hat{A} = S^{-1}AS$$

$$\hat{B} = S^{-1}BR$$

$$\hat{H} = HS$$

$$S \in Gl(n), \quad R \in O(m)$$

und die Matrizen $\hat{A}, \hat{B}$ und $\hat{H}$ haben die in

Bild 3 gezeigte Struktur, wobei $k \geq 1$, $t_i > 0$

für $i \in \underline{k}$, $0 < s_i \leq t_i$ für $i \in \underline{k-1}$, $0 \leq s_k \leq t_k$,

$t \geq 0$, $\text{rg } E_{i-1} = t_i$ für $i \in \underline{k}$ und

$D_i \in Gl(t_i - s_i)$ für $i \in \underline{k}$ (und $t_i - s_i \geq 0$).

$$(ii) \quad V^1(\hat{A}, \text{im } \hat{B}, \ker \hat{H}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{matrix} t_1 + t_2 + \dots + t_i \end{matrix} \quad \text{für } i \in \underline{k}.$$

$$(iii) \quad V^\infty(\hat{A}, \text{im } \hat{B}, \ker \hat{H}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{matrix} t \\ t_1 + t_2 + \dots + t_k \end{matrix}.$$

Beweis

(i): Für jede Stop-Bedingung prüft man die Behauptung leicht nach.

(ii): Der Beweis wird durch Induktion geführt. Für $i = 1$ gilt

$$v^1(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) = \ker \hat{A} = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix}_{t_1}.$$

Die Behauptung sei jetzt richtig für ein festes i , $i < k$,
d.h. es sei

$$v^i = v^i(\hat{A}, \text{im } \hat{B}, \ker \hat{A}) = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix}_{t_1+t_2+\dots+t_i}.$$

Ferner sei

$$x = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}_{t_1+t_2+\dots+t_{i+1}} \in \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix}_{t_1+t_2+\dots+t_{i+1}}.$$

Dann gilt einerseits $\hat{A}x = 0$ und andererseits

$$\hat{A}x = [* \quad \underbrace{* \quad 0}_{t_i} \quad \underbrace{* \quad 0}_{t_{i-1}} \quad \dots \quad \underbrace{* \quad 0}_{t_2} \quad \underbrace{* \quad 0}_{t_1}]^T \in v^i + \text{im } \hat{B},$$

$\begin{matrix} s_i & s_{i-1} & & s_2 & s_1 \\ & & & & \end{matrix}$

d.h. $x \in v^{i+1}$. Andersherum sei

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t_1+t_2+\dots+t_{i+1}} \in v^{i+1} = \ker \hat{A} + \hat{A}^{-1}(v^i + \text{im } \hat{B}).$$

Ferner sei $\hat{L}$ diejenige $(s_1+s_2+\dots+s_i) \times (t_1+t_2+\dots+t_{i+1})$ -
Teilmatrix von $\hat{A}$, die die mit $s_1, s_2, \dots, s_i$ bezeichneten
Zeilen und die mit $t_1, t_2, \dots, t_{i+1}$ bezeichneten Spalten
von $\hat{A}$ enthält. Ferner sei die $\ell \times (t_1+t_2+\dots+t_{i+1})$ -Matrix
 $\hat{M}$ definiert durch $\hat{A} = [0, \hat{M}]$. Man beachte, daß die
Matrix $\begin{bmatrix} \hat{L} \\ \hat{M} \end{bmatrix}$ vollen Spaltenrang besitzt, aus $\hat{L}x_2 = 0$ und
 $\hat{M}x_2 = 0$ also $x_2 = 0$ folgt. Dieser Schluß läßt sich in
der Tat anwenden, da $x \in \ker \hat{A}$ die Gleichung $\hat{M}x_2 = 0$
impliziert und

$$\hat{A}x \in v^i + \text{im } \hat{B} \Rightarrow \hat{L}x_2 = 0.$$

(iii): Wegen (ii) ist nur noch $V^{k+1} \supset V^k = \text{im} \begin{bmatrix} I \\ 0 \end{bmatrix} \{t_1 + t_2 + \dots + t_k\}$ zu zeigen. Dieses folgt aber genauso wie in (ii). $\square$

Bemerkung 7

Die Form aus Bild 3 entspricht der Form aus Lemma 2: es gilt $t = r_1$ und die letzten $n - r_1$ Zeilen der Form in Lemma 2 sind in Bild 3 "vermischt". Die Lemma 2(v) entsprechende Eigenschaft ist in Bild 3 aufgrund der zusätzlichen Struktur offensichtlich.

Bemerkung 8

Im Algorithmus MASIU werden so weit wie möglich orthogonale Transformationen durchgeführt. Lediglich Schritt 18 enthält nicht-orthogonale Operationen. Man beachte, daß es i.a. unmöglich ist, die Form aus Bild 3 durch orthogonale Transformationen zu erzeugen. Um dieses zu zeigen, sei $n = 3$, $\ell = 1$, $m = 2$ und

$$A = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix},$$

$$H = [0 \ 0 \ 1].$$

Für dieses System gilt $(\text{im } B)^\perp \cap \ker C = \{0\}$ und daher auch $(\text{im}(Q^T B R))^\perp \cap \ker(CQ) = \{0\}$ für alle $Q \in O(3)$ und $R \in O(2)$.

Durch den Algorithmus MASIU wird aber das folgende System mit $(\text{im } \hat{B})^\perp \cap \ker \hat{C} \neq \{0\}$ erzeugt:

$$\hat{A} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix},$$

$$\hat{H} = [0 \ 0 \ 1].$$

Im folgenden soll der Algorithmus MASIU zur Lösung von FSEPSSZ herangezogen werden. Nimmt man Lösbarkeit des Problems an, so liefert der Algorithmus, angewendet auf A^T, H^T, B^T , (bei simultaner Transformation G), eine Matrix $S \in Gl(n)$, so daß das System $\hat{A} = S^{-1}AS, \hat{B} = S^{-1}B, \hat{G} = S^{-1}G, \hat{H} = HS$ die in Bild 4 dargestellte Form hat, wobei E_i für $i = 0, 1, 2, \dots, k-1$ vollen Zeilenrang besitzt und $(\hat{A}, \hat{B})$ erreichbar ist.

Der folgende Satz wird jetzt konstruktiv bewiesen.

Satz 6

Es sei ein erreichbares System $\hat{A}, \hat{B}, \hat{G}, \hat{H}$ der in Bild 4 dargestellten Form, wobei $k \geq 1, t_i > 0$ für $i \in \underline{k}$, $0 < s_i \leq t_i$ für $i \in \underline{k-1}, 0 \leq s_k \leq t_k, t \geq 0$ und $\text{rg } E_{i-1} = t_i$ für $i \in \underline{k}$, gegeben. Für dieses System ist FSEPSSZ lösbar.

Die einzelnen Schritte des Beweises sind in den folgenden drei Lemmata zusammengefaßt.

Lemma 4

Es sei ein System $\hat{A}, \hat{B}, \hat{G}, \hat{H}$ mit den Eigenschaften aus Satz 6 und $M \in \mathbb{R}$ gegeben. Partitioniere das System wie folgt:

$$\hat{A} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \underbrace{\hat{A}_{21}}_t & \hat{A}_{22} \end{bmatrix} t, \quad \hat{B} = \begin{bmatrix} 0 \\ \hat{B}_2 \end{bmatrix} t, \quad \hat{G} = \begin{bmatrix} 0 \\ \hat{G}_2 \end{bmatrix} t, \quad (29)$$

$$\hat{H} = \begin{bmatrix} \hat{H}_1 \\ \hat{H}_2 \end{bmatrix} t.$$

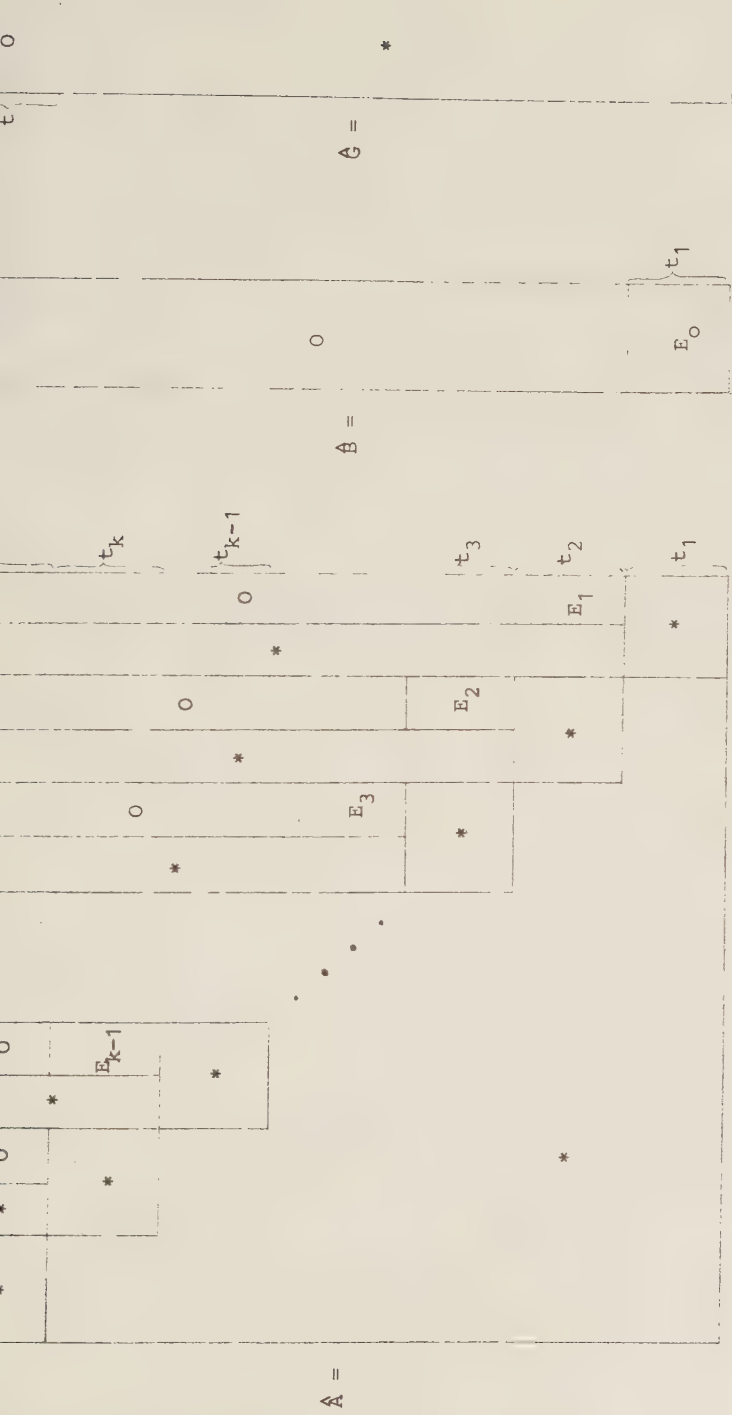


Bild 4



Dann gibt es eine Ähnlichkeitstransformation $\hat{S} \in \text{Gl}(n)$ und eine Rückführung $\hat{F} \in \mathbb{R}^{m \times n}$, so daß

$$\tilde{A}: = \hat{S}^{-1} \hat{A} \hat{S} + \hat{S}^{-1} \hat{B} \hat{F} = \begin{bmatrix} \tilde{A}_{11} & \hat{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}, \quad \tilde{B}: = \hat{S}^{-1} \hat{B} = \hat{B}, \quad \tilde{C}: = \hat{S}^{-1} \hat{C} = \hat{C}, \quad (30)$$

$$\tilde{H}: = \hat{H} \hat{S} = [\tilde{H}_1 \quad \hat{H}_2],$$

wobei $\tilde{A}_{22}$ die gleiche Struktur wie $\hat{A}_{22}$ in (29) besitzt und $\text{spec}(\tilde{A}_{11}) \subset \mathbb{C}_M := \{s \in \mathbb{C} \mid \text{re}(s) < M\}$.

Beweis

Da $(\hat{A}, \hat{B})$ erreichbar ist, ist auch $(\hat{A}_{11}, \hat{A}_{12})$ erreichbar und es existiert $K \in \mathbb{R}^{(t_1+t_2+\dots+t_k) \times t}$, so daß $\text{spec}(\tilde{A}_{11}) \subset \mathbb{C}_M$, wobei $\tilde{A}_{11} := \hat{A}_{11} + \hat{A}_{12} K$. Definiert man nun $S': = \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}$, und $A': = (S')^{-1} \hat{A} S'$, $B': = (S')^{-1} \hat{B}$, $G': = (S')^{-1} \hat{C}$, $H': = \hat{H} S'$, so folgt

$$A' = \begin{bmatrix} \tilde{A}_{11} & \hat{A}_{12} \\ A'_{21} & A'_{22} \end{bmatrix}, \quad B' = \hat{B}, \quad G' = \hat{C},$$

$$H' = [H'_1 \quad \hat{H}_2],$$

wobei A'_{22} die gleiche Struktur wie $\hat{A}_{22}$ besitzt.

Da die E_i vollen Zeilenrang besitzen und "über" den E_i nur Nullen sind, gibt es eine weitere Ähnlichkeitstransformation $S'' \in \text{Gl}(n)$ der Form $S'' = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} t$, so daß für das System $A'': = (S'')^{-1} A' S''$, $B'': = (S'')^{-1} B'$, $G'': = (S'')^{-1} G'$, $H'': = H' S''$ gilt:

$$A'' = \begin{bmatrix} \tilde{A}_{11} & \hat{A}_{12} \\ A''_{21} & A''_{22} \end{bmatrix}, \quad B'' = \hat{B}, \quad G'' = \hat{G},$$

$$H'' = [H''_1 \quad \hat{H}_2],$$

wobei A''_{22} die gleiche Struktur wie $\hat{A}_{22}$ besitzt und

$$A''_{21} = \begin{bmatrix} 0 \\ * \end{bmatrix}_{t_1}. \quad \text{Durch eine Zustandsrückführung läßt sich}$$

nun A''_{21} gänzlich auslöschen.

□

Das folgende Lemma, zusammen mit Lemma 4, besagt, daß man sich auf Systeme mit $t = 0$ beschränken kann.

Lemma 5

Es sei

$$A = \begin{bmatrix} \underbrace{A_{11}}_{n_1} & A_{12} \\ 0 & \underbrace{A_{22}}_{n_2} \end{bmatrix}_{n_1+n_2}, \quad G = \begin{bmatrix} 0 \\ G_2 \end{bmatrix}_{n_2}, \quad H = \begin{bmatrix} \underbrace{H_1}_{n_1} & \underbrace{H_2}_{n_2} \end{bmatrix},$$

wobei $\text{spec}(A) \subset \mathbb{C}_0$. Dann gilt

$$\|He^{A(\cdot)}G\|_1 \leq \|H_1e^{A_{11}(\cdot)}\|_1 \cdot \|A_{12}e^{A_{22}(\cdot)}G_2\|_1 + \|H_2e^{A_{22}(\cdot)}G_2\|_1.$$

Beweis

Es sei e_i der i -te Einheitsvektor im $\mathbb{R}^{n_1+n_2}$ und

$$e_i = \begin{bmatrix} e_{i1} \\ e_{i2} \end{bmatrix}_{n_1+n_2}. \quad \text{Dann ist die } i\text{-te Spalte von } e^{A(\cdot)}$$

Lösung der Differentialgleichung

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} e_{i1} \\ e_{i2} \end{bmatrix}.$$

Insbesondere gilt $\dot{x}_2 = A_{22}x_2$, $x_2(0) = e_{i2}$, d.h.

$x_2(t) = e^{A_{22}t} e_{i2}$. Ferner gilt $\dot{x}_1 = A_{11}x_1 + A_{12}x_2$, $x_1(0) = e_{i1}$, d.h.

$$x_1(t) = e^{A_{11}t} e_{i1} + ((e^{A_{11}(\cdot)}) * (A_{12}x_2(\cdot)))(t).$$

Insgesamt gilt also

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} e^{A_{11}t} & ((e^{A_{11}(\cdot)}) * (A_{12}e^{A_{22}(\cdot)}))(t) \\ 0 & e^{A_{22}t} \end{bmatrix} e_i,$$

d.h.

$$He^{At}G = ((H_1e^{A_{11}(\cdot)}) * (A_{12}e^{A_{22}(\cdot)}G_2))(t) + H_2e^{A_{22}(\cdot)}G_2.$$

□

Das folgende Lemma behandelt den Fall $t = 0$.

Lemma 6

Es sei ein System $\hat{A}$, $\hat{B}$, $\hat{G}$, $\hat{H}$ mit den Eigenschaften aus Satz 6 gegeben, wobei zusätzlich $t = 0$, und $\hat{A}$ wie folgt partitioniert ist:

$$\hat{A} = \begin{bmatrix} \hat{A}_1 \\ \hat{A}_2 \\ \vdots \\ \hat{A}_r \end{bmatrix}.$$

Für alle $M \in \mathbb{R}$ und $\bar{\varepsilon} > 0$ gibt es dann eine Rückführung $\hat{F} \in \mathbb{R}^{m \times n}$ mit

$$\| \hat{H}_i e^{(\hat{A} + \hat{B}\hat{F})(\cdot)} \hat{G}_1 \|_1 < \bar{\epsilon} \quad \text{für alle } i \in \underline{r},$$

$$\text{spec}(\hat{A} + \hat{B}\hat{F}) \subset \mathbb{C}_M.$$

Beweis (Induktion)

Für $k = 1$ hat $\hat{B}$ vollen Zeilenrang und die Behauptung ist offensichtlich. Die Behauptung sei jetzt richtig für aus $k-1$ Teilsystemen bestehende Systeme. Ein System mit k Teilsystemen kann man dann wie folgt partitionieren:

$$\hat{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} \} t_k \\ \} t_1 + t_2 + \dots + t_{k-1} \\ \} t_k \quad t_{1+1} \quad t_2 + \dots + t_{k+1} \end{matrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \quad \hat{G} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix},$$

$$H_i = [H_{i1} \quad H_{i2}].$$

Da die E_i vollen Zeilenrang besitzen, kann man durch eine Ähnlichkeitstransformation $S' \in GL(n)$ der Form

$$S' = \begin{bmatrix} I & 0 \\ * & I \end{bmatrix} \begin{matrix} \} t_k \\ \} t_1 + \dots + t_{k-1} \end{matrix} \quad \text{und eine Rückführung } F' \in \mathbb{R}^{m \times n} \text{ erreichen,}$$

daß die ersten t_k Spalten von $\hat{A}$ beliebige Werte annehmen. Speziell kann man die ersten Spalten wie folgt wählen:

$$\begin{bmatrix} A'_{11} \\ 0 \end{bmatrix} \begin{matrix} \} t_k \\ \} t_1 + \dots + t_{k-1} \\ \} t_k \end{matrix},$$

wobei

$$\text{spec}(A'_{11}) \subset \mathbb{C}_M$$

$$\| H_{i1} e^{A'_{11}(\cdot)} G_1 \|_1 < \bar{\epsilon}/3, \quad i \in \underline{r},$$

$$\|_{H_{11}} e^{A'_{11}(\cdot)}\|_1 < 1, \quad i \in \underline{r}.$$

Aufgrund der kompatiblen Struktur von $\hat{A}$ kann man S' zusätzlich so wählen, daß die H_{11} , $i \in \underline{r}$, durch die Ähnlichkeitstransformationen nicht verändert werden.

Das entstehende System $A': = (S')^{-1}\hat{A}S' + (S')^{-1}\hat{B}F'$,
 $B': = (S')^{-1}\hat{B}$, $G': = (S')^{-1}\hat{G}$, $H'_i = \hat{H}_i S'$ hat dann die Form

$$A' = \begin{bmatrix} A'_{11} & A'_{12} \\ 0 & A'_{22} \end{bmatrix}, \quad B' = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \quad G' = \begin{bmatrix} G_1 \\ G'_2 \end{bmatrix},$$

$$H'_i = [H_{11} \quad H_{12}].$$

Man beachte, daß die Matrix A_{22} sich zwar ändert, die Struktur aber erhalten bleibt. Daher kann man aufgrund der Induktionsannahme eine Rückführung $F'' \in R^{\max(t_1+t_2+\dots+t_{k-1})}$ finden, so daß für $A''_{22} := A'_{22} + B_2 F''$ gilt:

$$\text{spec}(A''_{22}) \subset \mathbb{C}_M,$$

$$\|_{H_{12}} e^{A''_{22}(\cdot)} G'_2\|_1 \leq \bar{\epsilon}/3, \quad i \in \underline{r},$$

$$\|_{A_{12}} e^{A''_{22}(\cdot)} G'_2\|_1 \leq \bar{\epsilon}/3.$$

Insgesamt gilt dann mit

$$\hat{A} := (F' + [0 \ F'']) (S')^{-1}$$

einerseits $\text{spec}(\hat{A} + \hat{B}F) \subset \mathbb{C}_M$ und andererseits

$$\hat{A}_i e^{(\hat{A}+\hat{B}\hat{F})t} \hat{G} = H_i' e^{(A'+B'[0 \ F''])t} G,$$

$$= [H_{i1} \ H_{i2}] \begin{bmatrix} e^{A'_{11}t} & (e^{A'_{11}(\cdot)} * (A_{12} e^{A''_{22}(\cdot)})) (t) \\ 0 & e^{A''_{22}t} \end{bmatrix} \begin{bmatrix} G_1 \\ G'_2 \end{bmatrix}.$$

Also

$$\begin{aligned} & \hat{A}_i e^{(\hat{A}+\hat{B}\hat{F})(\cdot)} \hat{G} \|_1 \\ & \leq \|H_{i1} e^{A'_{11}(\cdot)} G_1\|_1 + \|H_{i1} e^{A'_{11}(\cdot)}\|_1 \cdot \|A_{12} e^{A''_{22}(\cdot)} G'_2\|_1 \\ & \quad + \|H_{i2} e^{A''_{22}(\cdot)} G'_2\|_1 \\ & \leq \bar{\varepsilon}/3 + 1 \cdot \bar{\varepsilon}/3 + \bar{\varepsilon}/3 = \bar{\varepsilon}. \end{aligned}$$

□

Mit diesen Vorbereitungen läßt sich der Beweis von Satz 6 einfach führen:

Beweis von Satz 6

Es sei $M \in \mathbb{R}$, $\varepsilon > 0$ und $d \in L_p((0, \infty), \mathbb{R}^q)$ gegeben. Wegen

$$\|z(\cdot)\|_p \leq \|H e^{(A+BF)(\cdot)} G\|_1 \cdot \|d\|_p$$

ist zu zeigen, daß es ein $F \in \mathbb{R}^{m \times n}$ gibt, so daß

$\text{spec}(A+BF) \subset \mathbb{C}_M$ und $\|H e^{(A+BF)(\cdot)} G\|_1 \leq \varepsilon$. Wählt man $\hat{A}$

und $\hat{F}$ nach Lemma 4, so ist aufgrund von Lemma 5 nur noch zu

zeigen, daß ein $F_2 \in \mathbb{R}^{m \times (t_1+t_2+\dots+t_k)}$ existiert mit

$$\text{spec}(\tilde{A}_{22} + \hat{B}_2 F_2) \subset \mathbb{C}_m$$

$$\| \hat{A}_{12} e^{(\tilde{A}_{22} + \hat{B}_2 F_2)(\cdot)} \hat{G}_2 \|_1 \leq 1/(2 \| \tilde{H}_1 e^{\tilde{A}_{11}(\cdot)} \|_1)$$

$$\| \hat{A}_2 e^{(\hat{A}_{22} + \hat{B}_2 F_2)(\cdot)} \hat{G}_2 \| \leq 1/2 .$$

Da $\| \tilde{H}_1 e^{\tilde{A}_{11}(\cdot)} \|_1$ endlich ist, folgt dieses aus Lemma 6.

Bemerkung 9

Satz 6 zusammen mit Algorithmus MASIU gibt einen zu (Willems 1981) alternativen Beweis der einen Implikation von Satz 5. Dieser Beweis hat den Vorteil, daß er das Problem nicht auf ein skalares reduziert, und daß sich die Konstruktion direkt in einen Matrix-Algorithmus zur Berechnung des Reglers übersetzen läßt. So erhält man auch eine "Zweischrittmethode" zur Lösung von FSEPSSZ. Allerdings ist der zugehörige Algorithmus nicht numerisch stabil.

Bemerkung 10

Anstatt den Algorithmus MAIU zum Algorithmus MASIU zu modifizieren, hätte man auch einen auf den Algorithmus MAIU aufbauenden Algorithmus zur Berechnung der Zustandsdarstellungsform aus Bild 3 entwerfen können. Man hätte dann das System aus Lemma 2 sukzessive in das System aus Bild 3 überführt. Da dabei aber viele Rechnungen doppelt auszuführen sind, erscheint die hier vorgestellte Methode sowohl vom Gesichtspunkt der numerischen Stabilität, als auch vom Gesichtspunkt des Rechenaufwands vorteilhafter zu sein. Ein Vorteil der

anderen Methode ist allerdings, daß die Lösbarkeit während des Algorithmus numerisch stabil überprüft werden kann.

Die oben diskutierte Methode zur Lösung von FSEPSSZ ist sehr aufwendig. Es gibt jedoch zwei wichtige Spezialfälle, für die sich die Rechnungen wesentlich vereinfachen lassen.

Spezialfall 1: $m = 1$

Falls FSEPSSZ lösbar ist, liefert in diesem Fall der Algorithmus MAIU die folgende Zustandsdarstellungsform:

$$\hat{A} = \begin{bmatrix} * & * & & 0 \\ \hline * & * & \oplus & 0 \\ \hline & & * & \oplus \\ & & & \ddots \\ * & * & & * \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \oplus \end{bmatrix}, \quad \hat{G} = \begin{bmatrix} 0 \\ \hline G_2 \\ \hline G_3 \end{bmatrix}, \quad (31)$$

$$\hat{A} = \begin{bmatrix} H_1 & H_2 & 0 \end{bmatrix}.$$

Dieses ist bereits das System aus Bild 4, und daher läßt sich der numerisch instabile Algorithmus MASIU vermeiden. Die Form (31) läßt sich sogar herstellen, indem man (A,B) in System-Hessenberg-Form transformiert. Daher reicht in diesem Fall bereits der Algorithmus SHF aus Kapitel 3 aus. Die dadurch entstehende Form (31) wurde in (Linnemann 1984) zur Charakterisierung der Lösbarkeit eines dezentralen Regelungsproblems, bei dem die Teilsysteme skalare Eingänge haben, herangezogen. Dieses Resultat läßt sich mittels der

Zustandsdarstellungsform aus Bild 4 sofort auf multi-variable Teilsysteme verallgemeinern.

Spezialfall 2: $\text{im } G \subset \text{im } B$

Falls $\text{im } G \subset \text{im } B$ und (A, B) erreichbar ist, so ist FSEPSSZ lösbar. Für diesen Spezialfall läßt sich der Regler durch den folgenden einfachen Algorithmus bestimmen.

Algorithmus 12

1. Eingabe $n, m, q, l \geq 0$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $G \in \mathbb{R}^{n \times q}$, $H \in \mathbb{R}^{l \times n}$, $M \in \mathbb{R}$, $\epsilon > 0$.

2. Schiebe die Zeilen von B nach unten mit $Q^T \in O(n)$:

$$Q^T B = \begin{bmatrix} 0 \\ B_2 \end{bmatrix} \}_{r},$$

wobei B_2 vollen Zeilenrang hat.

3. Berechne

$$\tilde{A} = Q^T A Q$$

$$\tilde{B} = Q^T B$$

$$\tilde{G} = Q^T G$$

$$\tilde{H} = H Q$$

4. Kommentar: Es gilt

$$\tilde{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & \underbrace{A_{22}}_r \end{bmatrix} \}_{r}, \quad \tilde{B} = \begin{bmatrix} 0 \\ B_2 \end{bmatrix}, \quad \tilde{G} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \}_{r},$$

$$\tilde{H} = [H_1 \quad \underbrace{H_2}_r].$$

5. Falls $G_1 \neq 0$, so stoppe (im $G \nsubseteq \text{im } B$).

6. Falls (A_{11}, A_{12}) nicht erreichbar ist, so stoppe
 $((A, B)$ ist nicht erreichbar).

7. Bilde

$$S = \begin{bmatrix} I & O \\ K & I \end{bmatrix} \in \text{Gl}(n),$$

wobei K so gewählt ist, daß $\text{spec}(A_{11} + A_{12}K) \subset \mathbb{C}_M$.

8. Berechne

$$\bar{A} = S^{-1} \tilde{A} S$$

$$\bar{B} = S^{-1} \tilde{B}$$

$$\bar{G} = S^{-1} \tilde{G}$$

$$\bar{H} = \tilde{H} S.$$

9. Kommentar: Es gilt

$$\bar{A} = \begin{bmatrix} A_{11} + A_{12}K & A_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} O \\ B_2 \end{bmatrix}, \quad \bar{G} = \begin{bmatrix} O \\ G_2 \end{bmatrix},$$

$$\bar{H} = \begin{bmatrix} \bar{H}_1 & H_2 \end{bmatrix}.$$

10. Berechne $\bar{F} \in \mathbb{R}^{m \times n}$ so, daß

$$\bar{A} + \bar{B}\bar{F} = \begin{bmatrix} A_{11} + A_{12}K & A_{12} \\ O & A'_{22} \end{bmatrix},$$

wobei $\text{spec}(A'_{22}) \subset \mathbb{C}_M$ und

$$\|H_2 e^{A'_{22}(\cdot)} G_2\|_1 \leq \varepsilon/2,$$

$$\|A_{12} e^{A'_{22}(\cdot)} G_2\|_1 \leq \varepsilon/(2 \|\bar{H}_1 e^{(A_{11} + A_{12}K)(\cdot)}\|_1).$$

11. Berechne $F = \bar{F} S^{-1} Q^T$ und stoppe.

Bemerkung 11

Der in Schritt 11 berechnete Regler löst FSEPSSZ, da

$$\begin{aligned} \|H e^{A(\cdot)G}\|_1 &\leq \|H_1 e^{(A_{11}+A_{12}K)(\cdot)}\|_1 \cdot \|A_{12} e^{A'_{22}(\cdot)}\|_1 \|G_{21}\|_1 \\ &\quad + \|H_2 e^{A'_{22}(\cdot)}\|_1 \|G_{21}\|_1 \leq \epsilon. \end{aligned}$$

Bemerkung 12

Der Algorithmus 12, in Zusammenhang mit den Überlegungen in (Linnemann 1984), liefert insbesondere einen einfachen Beweis für die Tatsache, daß ein zusammengesetztes System, bei dem die Subsysteme nur durch ihre Ein- und Ausgänge verkoppelt sind, stets durch dezentrale Zustandsrückführung stabilisierbar ist. Der ursprüngliche Beweis hierfür (Sezer und Hüseyin 1978) ist wesentlich komplizierter und baut auf multivariable Frobenius-Formen (vgl. Kapitel 3) auf. Es steht außer Zweifel, daß der hier vorgeschlagene Algorithmus 12 numerisch attraktiver als der von Sezer und Hüseyin (1978) ist.

5.4. Matrizielle und strukturelle Charakterisierungen der Lösbarkeit

Für viele Regelungsprobleme sind geometrische Charakterisierungen der Lösbarkeit bekannt (vgl. z.B. Wonham 1974). Unter geometrischen Charakterisierungen werden in diesem Zusammenhang Charakterisierungen in Termen von Unterräumen, linearen Abbildungen usw. verstanden. In Abschnitt 1 dieses Kapitels sind

einige dieser Charakterisierungen wiedergegeben. Diese Charakterisierungen führen auch zu geometrischen Algorithmen, mit deren Hilfe sich die Probleme lösen lassen. Zur Durchführung auf einem Computer sind diese Algorithmen jedoch nicht geeignet. Dazu hat man von Unterräumen und linearen Abbildungen zu Basen und linearen Abbildungen überzugehen. In den vorhergehenden Abschnitten wurden entsprechende Matrix-Versionen der Algorithmen aus Abschnitt 5.1. vorgestellt. In diesem Abschnitt wird nun gezeigt, daß nicht nur die Algorithmen, sondern auch die Charakterisierungen der Lösbarkeit aus Abschnitt 5.1. matrizielle Versionen besitzen. Es wird sich dabei zeigen, daß viele der in den Abschnitten 5.2. und 5.3. betrachteten speziellen Zustandsdarstellungsformen bereits charakteristisch für das betreffende Störungsentkopplungsproblem sind.

Um den Unterschied zwischen einer geometrischen und einer matriziellen Charakterisierung der Lösbarkeit zu verdeutlichen, betrachte man die Eigenschaft der Erreichbarkeit eines Systems $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$:

geometrische Charakterisierung: $\langle A | \text{im } B \rangle = \mathbb{R}^n$;

matrizielle Charakterisierung: Es gibt $S \in \text{Gl}(n)$, so daß
 $(S^{-1}AS, S^{-1}B)$ ein
Hessenberg-System ist.

5.4.1. Störungsentkopplung durch Zustandsrückführung

Der folgende Satz ist eine matrizielle Version des Satzes 1.

Satz 7

SEPZ ist genau dann lösbar, wenn $S \in \text{Gl}(n)$ existiert mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} B_1 \\ B_2 \\ 0 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \\ 0 \end{bmatrix}, \quad (32)$$

$$HS = \begin{bmatrix} 0 & H_2 & H_3 \end{bmatrix},$$

wobei $A_{ij} \in \mathbb{R}^{r_i \times r_j}$, $B_i \in \mathbb{R}^{r_i \times m}$, $G_i \in \mathbb{R}^{r_i \times q}$, $H_i \in \mathbb{R}^{l \times r_j}$,
 $r_1, r_2, r_3 \geq 0$ und - falls $r_1 > 0$ und $r_2 > 0$ - $\text{rg } B_2 = r_2$.

Beweis

Die Hinlänglichkeit der Bedingung des Satzes wurde in Abschnitt 5.3.1. bereits konstruktiv gezeigt. Die Notwendigkeit ergibt sich aus Proposition 2(ii) und Satz 1. Sie läßt sich aber auch unabhängig vom Algorithmus MAIU wie folgt einsehen. Es sei $\{s_1, s_2, \dots, s_n\}$ eine Basis des $\mathbb{R}^n$, die mit der Kette

$$\{0\} \subset V_{\ker H}^* \subset V_{\ker H}^* + \text{im } B \subset \mathbb{R}^n$$

von Unterräumen verträglich ist, d.h. $s_1, \dots, s_{r_1}$ sei eine Basis von $V_{\ker H}^*$ und $s_1, \dots, s_{r_1+r_2}$ sei eine Basis von $V_{\ker H}^* + \text{im } B$. Bezüglich dieser Basis ergibt sich die Form (32).

□

Bemerkung 13

In (32) muß der durch die ersten r_1 Einheitsvektoren aufgespannte Raum nicht notwendig $V_{\ker H}^*$ sein; es kann irgendein (A,B) -invarianter Unterraum V mit $\text{im } G \subset V \subset \ker H$ sein.

Bemerkung 14

Die Notwendigkeit von (32) wurde mit Hilfe von Satz 1 bewiesen. Der Beweis läßt sich jedoch auch gänzlich unabhängig von Satz 1 mit Hilfe von Matrix-Argumenten führen.

Bemerkung 15

Die Matrix S in Satz 7 kann auch orthogonal gewählt werden, und die Bedingung (32) läßt sich mit Hilfe des numerisch stabilen Algorithmus MAIU überprüfen.

Bemerkung 16

Bild 5 zeigt das Signalflußdiagramm des Systems (32).

Daher läßt sich - stark vereinfacht - die Aussage von Satz 7 auch so formulieren: "SEPZ ist genau dann lösbar, wenn die Störung d und der Ausgang z durch den Eingang u getrennt werden", d.h. es muß eine nicht direkt bei d gelegene Stelle im Signalflußdiagramm geben, die jede Störung von d nach z passieren muß, und die durch u vollständig beeinflußt werden kann. Diese Stelle ist in Bild 5 durch einen * gekennzeichnet. Noch weiter vereinfacht kann man sagen: "SEPZ ist genau dann lösbar, wenn gilt: erst d , dann u , dann z ".¹⁾

¹⁾ Diese Vermutung hat Prof. J.C. Willems während seines Gastaufenthalts 1982 an der Universität Bremen geäußert und bildete den Ausgangspunkt für die Untersuchungen, die zu dem vorliegenden Abschnitt 5.4 geführt haben.

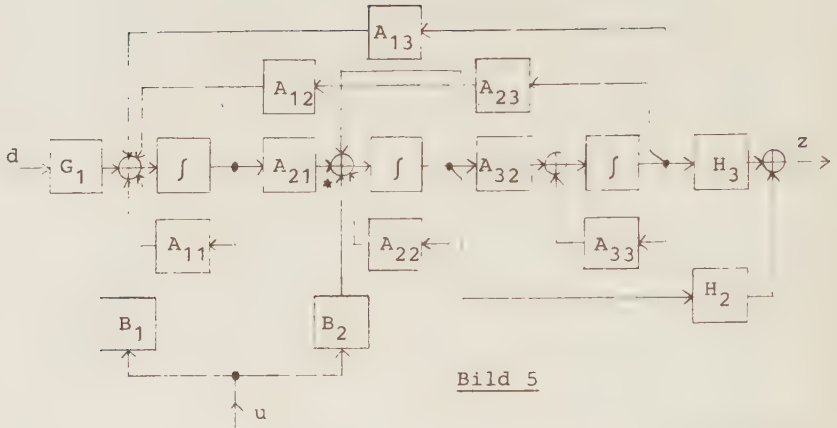


Bild 5

Faßt man die letzten beiden Teilsysteme in (32) zusammen, so erhält man:

Folgerung 1

SEPZ ist genau dann lösbar, wenn $S \in \text{Gl}(n)$ ($S \in \text{O}(n)$) existiert mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \end{bmatrix}, \quad (33)$$

$$HS = \begin{bmatrix} 0 & H_2 \end{bmatrix},$$

wobei $A_{ij} \in \mathbb{R}^{r_i \times r_j}$, $B_i \in \mathbb{R}^{r_i \times m}$, $G_1 \in \mathbb{R}^{r_1 \times q}$, $H_2 \in \mathbb{R}^{r_2 \times r_2}$, $r_1, r_2 \geq 0$ und $\text{im } A_{21} \subset \text{im } B_2$.

Da $\text{im } A_{21} \subset \text{im } B_2$ genau dann gilt, wenn eine Matrix $\tilde{A}_{21}$ mit $A_{21} = B_2 \tilde{A}_{21}$ existiert, folgt:

Folgerung 2

SEPZ ist genau dann lösbar, wenn $S \in \text{Gl}(n)$ ($S \in O(n)$) existiert mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} \\ B_2 \tilde{A}_{21} & A_{22} \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \end{bmatrix}, \quad (34)$$

$$HS = \begin{bmatrix} 0 & H_2 \end{bmatrix},$$

wobei $A_{ij} \in \mathbb{R}^{r_i \times r_j}$, $\tilde{A}_{21} \in \mathbb{R}^{m \times r_2}$, $B_i \in \mathbb{R}^{r_i \times m}$, $G_1 \in \mathbb{R}^{r_1 \times q}$, $H_2 \in \mathbb{R}^{\ell \times r_2}$ und $r_1, r_2 \geq 0$.

Das Signalflußdiagramm des Systems (34) ist in Bild 6 wiedergegeben.

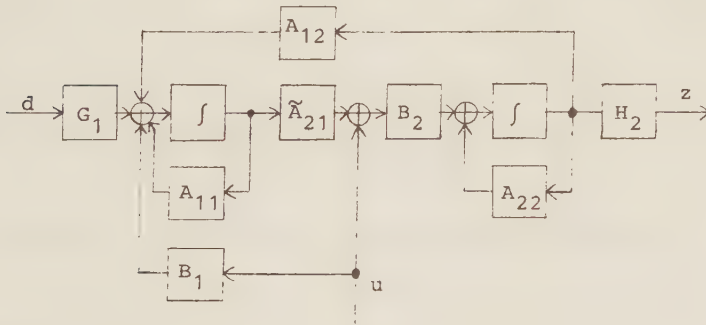


Bild 6

Aus Bild 6 ergibt sich unmittelbar die folgende strukturelle Charakterisierung von Bhattacharyya, Gomes und Howze (1983).

Folgerung 3

SEPZ ist genau dann lösbar, wenn das System die in Bild 7 wiedergegebene Struktur besitzt, wobei Σ_1 und Σ_2 lineare Systeme sind von der Form

$$\begin{aligned}\Sigma_1: \quad \dot{x}_1 &= A_1 x_1 + G_1 d + B_1 u + A_{12} v_3 \\ v_1 &= C_1 x_1\end{aligned}$$

$$\begin{aligned}\Sigma_2: \quad \dot{x}_2 &= A_2 x_2 + B_2 v_2 \\ z &= H_2 x_2 \\ v_3 &= D x_2,\end{aligned}$$

wobei $A_1 \in \mathbb{R}^{r_1 \times r_1}$, $G_1 \in \mathbb{R}^{r_1 \times q}$, $B_1 \in \mathbb{R}^{r_1 \times m}$, $A_{12} \in \mathbb{R}^{r_1 \times t}$,
 $t \geq 0$, $C_1 \in \mathbb{R}^{m \times r_1}$ und $A_2 \in \mathbb{R}^{r_2 \times r_2}$, $B_2 \in \mathbb{R}^{r_2 \times m}$,
 $H_2 \in \mathbb{R}^{l \times r_2}$, $D \in \mathbb{R}^{t \times r_2}$.

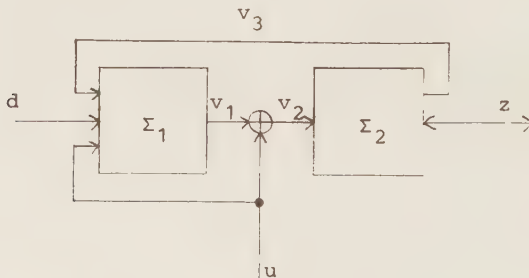


Bild 7

An Bild 7 wird die Bemerkung 16 besonders transparent.

Folgerung 3 wurde von *Bhattacharyya, Gomes und Howze (1983)* zur Ableitung eines Robustheitsresultats herangezogen.

Bezogen auf Folgerung 1 läßt sich dieses Resultat wie folgt formulieren.

Folgerung 4

Es sei SEPZ für das durch A, B, G, H gegebene System lösbar. Ferner sei S wie in Folgerung 1 gewählt und F wie in Abschnitt 5.3.1. berechnet. Das System sei jetzt zu $A + \delta A, B + \delta B, G + \delta G, H + \delta H$ gestört und $\delta A, \delta B, \delta G, \delta H$ seien wie das System in (33) partitioniert. Dann gilt: Falls $\delta A_{21} = 0, \delta B_2 = 0, \delta G_2 = 0$ und $\delta H_1 = 0$, so löst F auch das Störungsentkopplungsproblem für das gestörte Problem.

Man beachte, daß die Matrix S in Folgerung 1, und damit auch $\delta A_{21}, \delta B_2, \delta G_2$ und δH_1 , nicht eindeutig bestimmt sind.

In (Bhattacharyya, Gomes und Howze (1983) wird dieses Resultat so interpretiert, daß sich in Bild 7 wohl Σ_1 und Σ_2 , nicht aber die Struktur des Bildes ändern darf.

5.4.2. Störungsentkopplung durch Ausgangsrückführung

In diesem Abschnitt werden die Ergebnisse aus Abschnitt 5.4.1. auf SEPA übertragen.

Eine Matrix-Version von Satz 2 lautet wie folgt:

Satz 8

SEPA ist genau dann lösbar, wenn es $S \in \text{Gl}(n)$ gibt mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ 0 & A_{32} & A_{33} & A_{34} & A_{35} \\ 0 & A_{42} & A_{43} & A_{44} & A_{45} \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ 0 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ G_2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (34a)$$

$$CS = [0 \quad C_2 \quad C_3 \quad C_4 \quad C_5],$$

$$HS = [0 \quad 0 \quad 0 \quad H_4 \quad H_5],$$

wobei $A_{ij} \in \mathbb{R}^{i \times r_j}$, $B_i \in \mathbb{R}^{i \times m}$, $G_i \in \mathbb{R}^{i \times q}$, $C_i \in \mathbb{R}^{p \times r_i}$,

$H_i \in \mathbb{R}^{l \times r_i}$, $r_1, r_2, r_3, r_4, r_5 \geq 0$, $\text{rg } B_4 = r_4$ (falls $r_1 + r_2 + r_3 > 0$; $r_4 > 0$) und $\text{rg } C_2 = r_2$ (falls $r_2 > 0$; $r_3 + r_4 + r_5 > 0$).

Beweis

Die Hinlänglichkeit der Bedingung des Satzes wurde bereits in Abschnitt 5.3.2. konstruktiv nachgewiesen. Die Notwendigkeit ergibt sich aus der Behauptung in Schritt 8 von Algorithmus 9, kann aber auch unabhängig vom Algorithmus MAIU überprüft werden, indem man eine Basis des $\mathbb{R}^n$ wählt, die mit der Kette

$$\{0\} \subset S_{\text{im } G}^* \cap \ker C \subset S_{\text{im } G}^* \subset V_{\ker H}^* \subset V_{\ker H}^* + \text{im } B \subset \mathbb{R}^n$$

von Unterräumen verträglich ist. Bezüglich dieser Basis gilt nämlich (34a).

□

Die Bemerkungen 13 bis 15 gelten sinngemäß auch hier.

Bemerkung 17

An dem Signalflußdiagramm des Systems aus Satz 8 (vgl. Bild 1) kann man die folgende grobe Umschreibung von Satz 8 ablesen: "SEPA ist genau dann lösbar, wenn jede Störung, um von d nach z zu gelangen, zunächst eine Stelle im Signalflußdiagramm passieren muß, die *direkt* von y beobachtbar ist, und dann eine Stelle passieren muß, die u *direkt* beeinflussen kann". Noch weiter vereinfacht kann man sagen: "SEPA ist lösbar genau dann, wenn gilt: erst d , dann y , dann u , dann z ".

Die folgenden Resultate ergeben sich entsprechend den Folgerungen 1 bis 3.

Folgerung 5

SEPA ist genau dann lösbar, wenn $S \in \text{Gl}(n)$ existiert mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \\ 0 \end{bmatrix}, \quad (35)$$

$$CS = [C_1 \quad C_2 \quad C_3],$$

$$CH = [0 \quad 0 \quad H_3],$$

wobei $\text{im } A_{31} \subset \text{im } B_3$, $\text{im } A_{32} \subset \text{im } B_3$, $\ker C_1 \subset \ker A_{21}$ und $\ker C_1 \subset \ker A_{31}$.

Folgerung 6

SEPA ist genau dann lösbar, wenn $S \in \text{Gl}(n)$ existiert mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ \tilde{A}_{21}C_1 & A_{22} & A_{23} \\ B_3\tilde{A}_{31}C_1 & B_3\tilde{A}_{32} & A_{33} \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \\ 0 \end{bmatrix}, \quad (36)$$

$$CS = [\quad C_1 \quad \quad C_2 \quad C_3 \quad],$$

$$HS = [\quad 0 \quad \quad 0 \quad H_3 \quad].$$

Folgerung 7

SEPA ist genau dann lösbar, wenn das System die in Bild 8 wiedergegebene Struktur besitzt, wobei Σ_1 , Σ_2 und Σ_3 lineare Systeme sind von der Form

$$\begin{aligned} \Sigma_1: \quad & \dot{x}_1 = A_1x_1 + D_1v_5 + G_1d \\ & v_1 = C_1x_1 \\ \\ \Sigma_2: \quad & \dot{x}_2 = A_2x_2 + \tilde{A}_{23}v_6 + \tilde{A}_{21}v_1 + B_2u \\ & v_5 = A_{12}x_2 + A_{13}v_6 + B_1u \\ & v_2 = \tilde{A}_{32}x_2 + A_{31}v_1 \\ & v_4 = C_2x_2 + C_3v_6 \\ \\ \Sigma_3: \quad & \dot{x}_3 = A_3x_3 + B_3v_3 \\ & v_6 = D_2x_3 \\ & z = H_3x_3, \end{aligned}$$

wobei $x_1 \in \mathbb{R}^{r_1}$, $x_2 \in \mathbb{R}^{r_2}$, $x_3 \in \mathbb{R}^{r_3}$, $v_1 \in \mathbb{R}^p$, $v_2 \in \mathbb{R}^m$,
 $v_3 \in \mathbb{R}^m$, $v_4 \in \mathbb{R}^p$, $v_5 \in \mathbb{R}^{t_1}$, $v_6 \in \mathbb{R}^{t_2}$, $t_1 \geq 0$, $t_2 \geq 0$
 und die Matrizen von passendem Format sind.

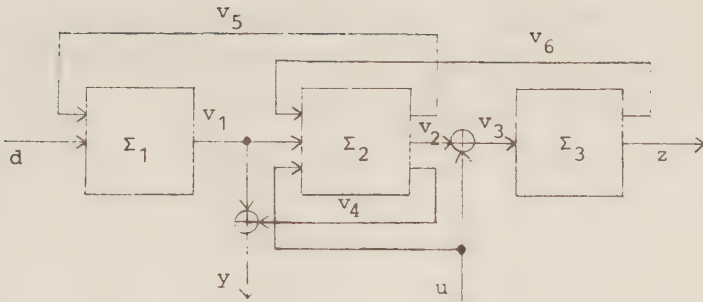


Bild 8

In Bild 8 sind Σ_1 , Σ_2 und Σ_3 also lineare Systeme, bei denen die Abhängigkeiten zwischen Ein- und Ausgängen teilweise kausal und teilweise strikt kausal sind.

Folgerung 7 ist eine Übertragung des Resultates von Bhattacharyya, Gomes und Howze (1983) auf Ausgangsrückführung. Auch das Robustheitsresultat aus Folgerung 4 läßt sich leicht auf Ausgangsrückführung übertragen. Auf eine explizite Formulierung wird hier verzichtet.

Das Bild 8 läßt sich auch im Frequenzbereich interpretieren. Um dieses zu präzisieren, wird SEPA zunächst im Frequenzbereich formuliert. Dazu seien die Gleichungen

$$\begin{aligned} z &= G_{zu}u + G_{zd}d \\ y &= G_{yu}u + G_{yd}d \end{aligned} \quad (37)$$

gegeben, wobei G_{zu} , G_{zd} , G_{yu} und G_{yd} strikt propre rationale Matrizen sind. Nach einer Rückführung der Form $u = Fy$, wobei F eine propre rationale Matrix ist, ergibt sich das geschlossene System

$$z = (G_{zu}F(I - G_{yu}F)^{-1}G_{yd} + G_{zd})d.$$

Also lautet das Problem im Frequenzbereich:

SEPA(F): Finde F (falls möglich), so daß

$$G_{zu}F(I - G_{yu}F)^{-1}G_{yd} + G_{zd} = 0.$$

Es gilt jetzt die folgende Charakterisierung der Lösbarkeit, ebenfalls im Frequenzbereich formuliert.

Folgerung 8

SEPA(F) ist genau dann lösbar, wenn sich die Gleichungen (37) schreiben lassen als

$$v_1 = G_1 \begin{bmatrix} v_5 \\ d \end{bmatrix}, \begin{bmatrix} v_5 \\ v_2 \\ v_4 \end{bmatrix} = G_2 \begin{bmatrix} v_6 \\ v_1 \\ u \end{bmatrix}, \begin{bmatrix} v_6 \\ z \end{bmatrix} = G_3 v_3, y = v_1 + v_4, v_3 = v_2 + u, \quad (38)$$

wobei G_1 , G_2 , G_3 propre rationale Matrizen sind und zusätzlich gilt:

- G_1 strikt proper;
- in G_2 ist die Abhängigkeit zwischen v_5 und v_1 , v_2 und v_6 , v_2 und u , v_4 und v_1 , v_4 und u strikt proper;
- G_3 strikt proper.

Beweis

Das System sei beschrieben durch die Gleichungen (38). Bildet man Realisierungen von G_1 , G_2 und G_3 , so erhält man Systeme der Form Σ_1 , Σ_2 , Σ_3 in Folgerung 7. Verkoppelt man diese Systeme wie in (38) beschrieben, so erhält man eine Realisierung des Systems (37) mit der Struktur wie in Bild 8. Nach Folgerung 7 ist SEPA für diese Realisierung lösbar, also ist auch SEPA(F) lösbar. Umgekehrt, falls SEPA(F) lösbar ist, so kann man Folgerung 7 auf eine beliebige Realisierung von (37) anwenden und man erhält (38).

■

5.4.3. Störungsentkopplung und Polzuweisung durch Zustandsrückführung

Eine matrizielle Version von Satz 3 lautet wie folgt:

Satz 9

SEPPZ ist genau dann lösbar, wenn (A,B) erreichbar ist, und $S \in \text{Gl}(n)$ und $R \in \text{Gl}(n)$ existieren mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ 0 & 0 & A_{43} & A_{44} \end{bmatrix}, \quad S^{-1}BR = \begin{bmatrix} B_{11} & 0 \\ 0 & 0 \\ 0 & B_{32} \\ 0 & 0 \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (39)$$

$$HS = \begin{bmatrix} 0 & 0 & H_3 & H_4 \end{bmatrix},$$

wobei $A_{ij} \in \mathbb{R}^{r_1 \times r_j}$, $B_{ij} \in \mathbb{R}^{r_1 \times m_j}$, $G_i \in \mathbb{R}^{r_1 \times q}$, $H_j \in \mathbb{R}^{l \times r_j}$,
 $r_1, r_2, r_3, r_4, m_1, m_2 \geq 0$, $m_2 = r_3$, $r_1 = 0$ falls $m_1 = 0$,
 und B_{32} invertierbar (falls $r_1 + r_2 > 0$ und $r_3 > 0$),

und (A_{11}, B_{11}) erreichbar (falls $m_1 > 0$ und $r_1 > 0$) ist.

Beweis

In Abschnitt 5.3.3. wurde gezeigt, daß sich jedes System, für das SEPPZ lösbar ist., in die Form (39) transformieren läßt (vgl. (23)). Andererseits ist leicht einzusehen, daß SEPPZ für das System in (39) lösbar ist.

Ein vom Algorithmus MAIU unabhängiger Beweis läßt sich führen, indem man zunächst eine Basis $y_1, y_2, \dots, y_k$ von $\text{im } B$ derart wählt, daß die ersten Vektoren $y_1, y_2, \dots, y_\ell$ ($0 \leq \ell \leq k$) eine Basis von $V_{\ker H}^* \cap \text{im } B$ bilden. Dann kann man eine Basis $\{x_1, \dots, x_n\}$ des $\mathbb{R}^n$ wie folgt konstruieren:

- $x_1, x_2, \dots, x_{r_1}$ sei eine Basis von $\langle A | V_{\ker H}^* \cap \text{im } B \rangle \cap V_{\ker H}^*$
 $= \langle A | \text{span}\{y_1, \dots, y_\ell\} \rangle \cap V_{\ker H}^*$

- $x_1, \dots, x_{r_1}, \dots, x_{r_1+r_2}$ sei eine Basis von $V_{\ker H}^*$

- $x_{r_1+r_2+i} = y_{\ell+i}$ für $i = 1, 2, \dots, k-\ell$ ($r_3 := k-\ell$)

- $x_1, x_2, \dots, x_n$ sei eine Basis des $\mathbb{R}^n$ ($r_4 := (n-r_1-r_2-r_3)$).

Zusätzlich kann man noch eine Basis $\{u_1, \dots, u_m\}$ des $\mathbb{R}^m$ so wählen, daß

- $B(\text{span}\{u_1, \dots, u_{m-r_3}\}) = \text{span}\{y_1, \dots, y_\ell\}$

- $B(\text{span}\{u_{m-r_3+1}, \dots, u_m\}) = \text{span}\{y_{\ell+1}, \dots, y_k\}$.

Bezüglich dieser Basen erhält man die Darstellung des Satzes.

□

Bemerkung 18

Bild 9 zeigt das Signalflußdiagramm des Systems (39). Daher läßt sich - stark vereinfacht - die Aussage von Satz 9 wie folgt formulieren: "SEPPZ ist genau dann lösbar, wenn jede

Störung, um von d nach z zu gelangen, zunächst ein erreichbares Teilsystem passieren muß und dann zu einer Stelle gelangt, die durch u *direkt* beeinflussbar ist".
 Noch weiter vereinfacht kann man sagen: "SEPPZ ist genau dann lösbar, wenn gilt: erst d und erreichbares Teilsystem, dann u , dann z ".

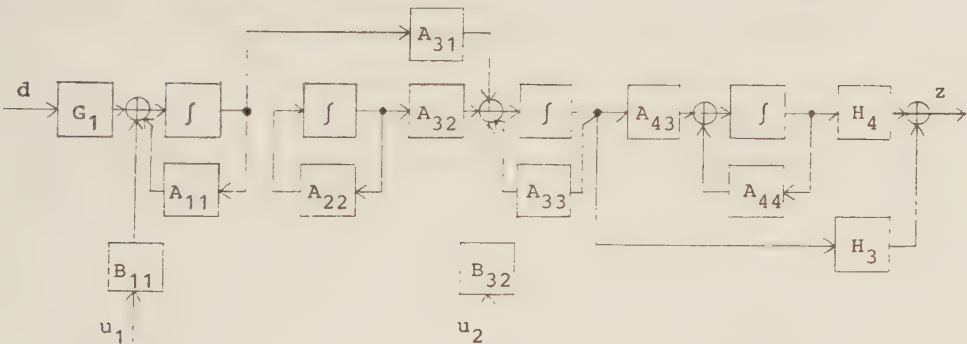


Bild 9 (die Kopplungen von rechts nach links sind nicht eingezeichnet)

Genauso wie in Abschnitt 5.4.1 erhält man aus Satz 9 einige Folgerungen:

Folgerung 9

SEPPZ ist genau dann lösbar, wenn (A,B) erreichbar ist und $S \in Gl(n)$ und $R \in Gl(m)$ existieren mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad S^{-1}BR = \begin{bmatrix} B_{11} & O \\ O & B_{22} \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ O \end{bmatrix}, \quad (40)$$

$$HS = \begin{bmatrix} O & H_2 \end{bmatrix},$$

wobei $\text{im } A_{21} \subset \text{im } B_{22}$ und (A_{11}, B_{11}) erreichbar ist.

Bemerkung 19

In Folgerung 9 und in der folgenden Folgerung 10 kann die Bedingung " (A, B) erreichbar" ersetzt werden durch " (A_{22}, B_{22}) erreichbar".

Folgerung 10

SEPPZ ist genau dann lösbar, wenn (A, B) erreichbar ist und $S \in \text{Gl}(n)$, $R \in \text{Gl}(m)$ existieren mit

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} \\ B_{22}\tilde{A}_{21} & A_{22} \end{bmatrix}, \quad S^{-1}BR = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix}, \quad S^{-1}G = \begin{bmatrix} G_1 \\ 0 \end{bmatrix},$$

$$HS = \begin{bmatrix} 0 & H_2 \end{bmatrix},$$

wobei (A_{11}, B_{11}) erreichbar ist.

Folgerung 11

SEPPZ ist genau dann lösbar, wenn das System die in Bild 10 wiedergegebene Struktur besitzt, wobei Σ_1 und Σ_2 lineare Systeme von der in Folgerung 3 beschriebenen Form sind, und zusätzlich (A_1, B_1) und (A_2, B_2) erreichbar sind.

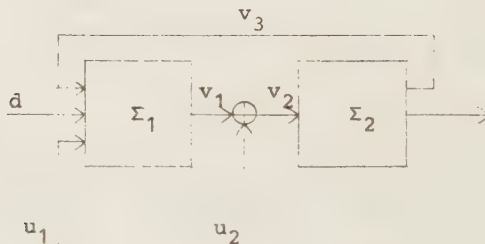


Bild 10

5.4.4. Fast Störungsentkopplung

Eine matrizielle Version von Satz 5 lautet wie folgt:

Satz 10

FSEPSZ ist genau dann lösbar, wenn $S \in \text{Gl}(n)$ existiert, so daß $\hat{A} := S^{-1}AS$, $\hat{B} := S^{-1}B$, $\hat{G} := S^{-1}G$ und $\hat{H} := HS$ die in Bild 4 dargestellte Struktur haben, wobei die in Satz 6 angeführten Eigenschaften gelten.

Beweis

Die eine Richtung wird durch Satz 6 abgedeckt. Die andere Richtung folgt aus Proposition 3(iii), Proposition 1 und Satz 5.

□

Bemerkung 20

Bemerkung 8 besagt, daß S i.a. nicht orthogonal gewählt werden kann.

Bemerkung 21

Für die Sätze 7, 8 und 9 wurde jeweils ein von Algorithmen unabhängiger Beweis gegeben. Ein derartiger Beweis ist für Satz 10 nicht offensichtlich.

In Bemerkung 10 wurde bereits erwähnt, daß anstatt des Algorithmus MASIU auch der Algorithmus MAIU zur Lösung von FSEPSZ benutzt werden kann. Dieses würde dann zu dem folgenden Resultat führen, welches hier nicht bewiesen wird.

ist in Bild 12 wiedergegeben. Dabei sind u , x_1 , x_2 und x_3 skalare Größen.

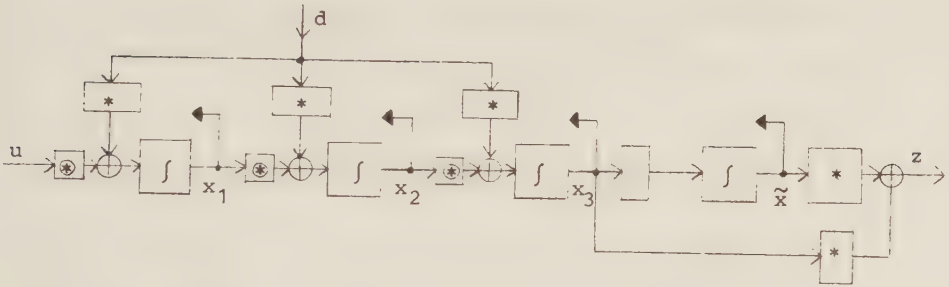


Bild 12

Grob gesprochen kann man daher sagen: "FSEPSSZ ist genau dann lösbar, wenn es eine Aufteilung der Zustandsvariablen in zwei Mengen X_1 und X_2 derart gibt, daß die Zustände in X_1 über u erreichbar sind, d die Zustände in X_2 nicht *direkt* beeinflusst und die Zustände in X_1 nicht *direkt* über z ausgelesen werden". Noch weiter vereinfacht läßt sich dieses so formulieren: "erst d und u , dann z ".

6. WAS IST FEEDBACK - WAS IST FEEDFORWARD ?

In der systemtheoretischen Literatur wird zwischen Feedback und Feedforward unterschieden. Es existieren jedoch noch keine präzisen Definitionen dieser Begriffe. Lediglich für den Fall eines Regelgesetzes der Form

$$u^k = Fx^k, \quad F \in \mathbb{R}^{m \times n}, \quad (1)$$

für ein System der Form¹⁾

$$x^{k+1} = Ax^k + Bu^k, \quad A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m} \quad (2)$$

hat Kalman (1972) eine Definition vorgeschlagen. Für feste A, B und F gibt er an, wann das Regelgesetz (1) Feedback bzw. Feedforward für das System (2) darstellt. Beispielsweise würde er für das System

$$\begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \vdots \\ x_6^{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1^k \\ x_2^k \\ \vdots \\ x_6^k \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1^k \\ u_2^k \end{bmatrix},$$

dessen Signalflußdiagramm in Bild 1 dargestellt ist, die "Rückführung" von x_2 nach u_1 Feedback und die "Rückführung" von x_5 nach u_1 Feedforward nennen.

¹⁾ In diesem Abschnitt werden zeitdiskrete Systeme betrachtet, da sich die Konzepte so besser motivieren lassen. Die Definitionen und Resultate lassen sich formal auch auf zeitkontinuierliche Systeme übertragen; die anschaulichen Interpretationen gehen dabei jedoch verloren.

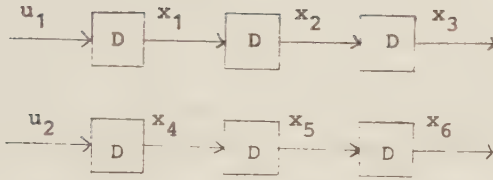


Bild 1 (D bezeichnet den Delayoperator $x_i^{k+1} \rightarrow x_i^k$)

Sein wesentliches Argument ist, daß bei der "Rückführung" von x_5 nach u_1 keine Schleife entsteht, was eine Voraussetzung für Feedback ist (vgl. Sprachgebrauch "feedback loop"). Die Arbeit von Kalman ist in (Dickinson 1976) weitergeführt.

Üblicherweise wird jedoch jedes Regelgesetz der Form (1) Feedback genannt. Im folgenden wird, als Alternative zu der Begriffsbildung von Kalman, eine Definition von Feedback und Feedforward vorgeschlagen, bei der insbesondere jedes Regelgesetz der Form (1) Feedback ist. Es wird sich dann herausstellen, daß jede störungsentkoppelnde Kontrolle Feedforward ist, bzw. mindestens Feedforward-Anteile besitzt.

Zur Motivation betrachte man zunächst das in Bild 2 dargestellte offene System der Form

$$x^{k+1} = Ax^k.$$

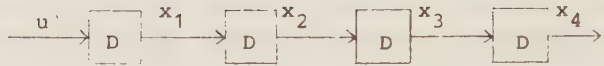


Bild 2: offenes System

Falls das geschlossene System die in Bild 3 dargestellte Form besitzt, so soll der dazugehörige Regler Feedback sein, während das geschlossene System in Bild 4 aus Bild 2 durch Feedforward hervorgeht.

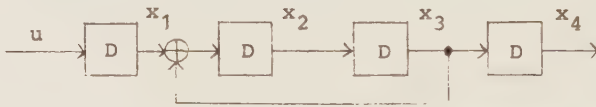


Bild 3: Feedback

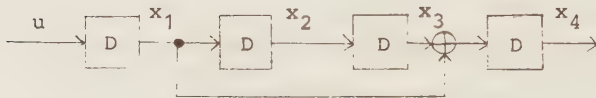


Bild 4: Feedforward

Eine Schwierigkeit offenbart jedoch Bild 5.

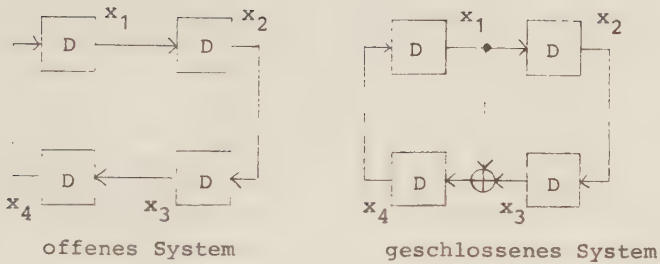


Bild 5: Feedback oder Feedforward ?

Das dort dargestellte geschlossene System kann man sich sowohl durch Feedback als auch durch Feedforward entstanden vorstellen: Betrachtet man den von x_3 ausgehenden Informationsfluß, so erhält man Feedback von x_3 über x_4 und x_1 zurück zum Summationspunkt. Andererseits, betrachtet man den von x_1 ausgehenden Informationsfluß, so erhält man Feedforward zum Summationspunkt, da x_2 "kurzgeschlossen" wird.

Die Schwierigkeit in dem obigen Beispiel liegt darin begründet, daß in dem System kein fester Bezugspunkt vorhanden ist. Wie das Beispiel zeigt, ist der Standpunkt, von dem aus man das System betrachtet, (entweder x_1 oder x_3) von entscheidender Bedeutung. Es wird daher nun angenommen, daß das System einen festen Bezugspunkt besitzt, der durch einen zusätzlichen Eingang d gegeben ist. Damit ist gemeint, daß jetzt Systeme der Form

$$\begin{aligned} x^{k+1} &= Ax^k + Bu^k + Gd^k \\ y^k &= Cx^k, \end{aligned} \quad (3)$$

wobei $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ und $G \in \mathbb{R}^{n \times q}$, betrachtet werden. Das Ziel dieses Abschnittes ist dann, zu definieren, wann ein Regler der Form

$$u^k = Fy^k + Kd^k, \quad (4)$$

wobei $F \in \mathbb{R}^{m \times p}$ und $K \in \mathbb{R}^{m \times q}$, Feedback bzw. Feedforward für das System (3) ist. Der Eingang d wird dabei der Bezugspunkt im System sein; er ist das "Fenster", durch das das System betrachtet wird. Wie in Kapitel 5 kann d eine Störung repräsentieren, es kann aber auch eine Referenzgröße, eine zusätzliche Kontrollgröße oder ein künstlich eingeführter Eingang zur Markierung eines kanonischen Bezugspunktes sein. Es wird zunächst angenommen, daß d skalar ist, d.h. $q = 1$.

Das durch (3) und (4) gegebene geschlossene System hat die Form

$$x^{k+1} = A_c x^k + G_c d^k, \quad (5)$$

wobei

$$\begin{aligned} A_c &= A + BFC \\ G_c &= G + BK, \end{aligned} \quad (6)$$

und Feedback bzw. Feedforward wird definiert durch Vergleich

der Signalflußdiagramme der Systeme (5) und

$$x^{k+1} = Ax^k + Gd^k. \quad (7)$$

Falls $(A, G) \in H(n, 1)$ (Bezeichnung wie in Kapitel 3), so hat das Signalflußdiagramm die in Bild 6 für $n = 3$ dargestellte Struktur.

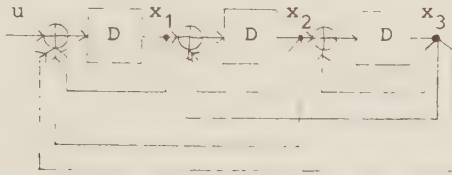


Bild 6

Man erkennt, daß die Zustände in der Reihenfolge $x_1, x_2, \dots, x_n$ erreicht werden, und daher prägt d den Zustandsvariablen eine "Ordnung" auf. Eine indefinite Situation wie in Bild 5 tritt nicht auf.

Man beachte, daß das Signalflußdiagramm eines Systems sehr stark von der gewählten Basis des Zustandsraumes abhängt. Die folgende Definition ist unabhängig von der Basis des Zustandsraumes, erhält aber andererseits die graphische Interpretation mittels Signalflußdiagrammen.

Definition 1

Es sei ein System (3) mit $q = 1$ und ein Regler (4) gegeben.

(a) Der Regler (4) heißt *Feedback* bezüglich (3), falls

$S \in Gl(n)$ existiert mit

$$S^{-1}_G = \begin{bmatrix} h_{10} \\ 0 \\ \vdots \\ 0 \\ \hline 0 \end{bmatrix}, \quad S^{-1}_{AS} = \begin{bmatrix} h_{11} & \cdot & \cdot & \cdot & \cdot & h_{1\ell} & & \\ h_{21} & \cdot & \cdot & \cdot & \cdot & \cdot & & H_{12} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ \hline & & & h_{\ell, \ell-1} & \cdot & h_{\ell\ell} & & \\ & & & 0 & & & & H_{22} \end{bmatrix}, \quad \begin{matrix} h_{i, i-1} \neq 0 \\ \text{für } i \in \underline{\ell}, \end{matrix} \quad (8)$$

und

$$S^{-1}_{BK} = \begin{bmatrix} 0 \\ \hline 0 \end{bmatrix}, \quad S^{-1}_{BFCS} = \begin{bmatrix} * & & & & & \\ & * & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & 0 & & & * & \\ \hline & & & 0 & & * \end{bmatrix}, \quad (9)$$

wobei $(S^{-1}_{BK}, S^{-1}_{BFCS}) \neq 0$.

(b) Der Regler (4) enthält *Feedback-Anteile* bezüglich (3), falls für alle $S \in \text{Gl}(n)$ mit (8) gilt:

$$M_2 \neq 0, \quad (10)$$

wobei M_1 und M_2 definiert sind durch

$$(S^{-1}_{BK}, S^{-1}_{BFCS}) = M_1 + M_2 \quad (11)$$

und

$$M_1 = \left(\begin{bmatrix} & & 0 & & & \\ & * & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ \hline & & & & & 0 \end{bmatrix} \right), \quad M_2 = \left(\begin{bmatrix} 0 & & & & & \\ & * & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & 0 & & & * & \\ \hline 0 & & & 0 & & * \end{bmatrix} \right). \quad (12)$$

- (c) Der Regler (4) heißt *Feedforward* bezüglich (3), falls $S \in \text{Gl}(n)$ existiert mit (8) und

$$S^{-1}_{BK} = \begin{bmatrix} & * \\ \text{---} & \\ & * \end{bmatrix}, \quad S^{-1}_{BFCS} = \begin{bmatrix} 0 & & & 0 & & 0 \\ & \ddots & & & & \\ & & * & & & \\ \text{---} & & & 0 & & \\ & & * & & & 0 \end{bmatrix}, \quad (13)$$

wobei $(S^{-1}_{BK}, S^{-1}_{BFCS}) \neq 0$.

- (d) Der Regler (4) enthält *Feedforward-Anteile* bezüglich (3), falls für alle $S \in \text{Gl}(n)$ mit (8) gilt:

$$M_1 \neq 0, \quad (14)$$

wobei M_1 und M_2 definiert sind durch die Gleichungen (11) und (12).

In den Gleichungen (8), (9), (12) und (13) ist die Aufteilung der Matrizen stets so zu verstehen, daß

$$\left[\begin{array}{c} \vdots \\ \text{---} \\ \vdots \end{array} \right] \begin{array}{l} \left. \begin{array}{c} \vdots \\ \vdots \end{array} \right\} \ell \\ \left. \begin{array}{c} \vdots \\ \vdots \end{array} \right\} n-\ell \end{array}, \quad \left[\begin{array}{cc} \vdots & \vdots \\ \text{---} & \text{---} \\ \vdots & \vdots \end{array} \right] \begin{array}{l} \left. \begin{array}{c} \vdots \\ \vdots \end{array} \right\} \ell \\ \left. \begin{array}{c} \vdots \\ \vdots \end{array} \right\} n-\ell \end{array}$$

Im folgenden wird die Bedeutung von Definition 1 diskutiert. Die folgende Bemerkung bezieht sich auf die zentrale Gleichung (8).

Bemerkung 1

Falls $S_0 \in \text{Gl}(n)$ das System (3) in die Form (8) transformiert, so ist die Menge aller Transformationsmatrizen, die das System in diese Form bringen, gegeben durch

$$\varphi := \{S = S_0 \cdot T \mid T \in \text{Gl}(n), T = \begin{bmatrix} * & & & \\ & * & & \\ & & \ddots & \\ 0 & & & * \\ \hline & & & 0 & \\ & & & & * \end{bmatrix}\}. \quad (15)$$

Die Menge φ enthält eine orthogonale Matrix, und die dazugehörige Form (8) kann numerisch stabil berechnet werden.

Beweis

Falls $S \in \varphi$, so ergibt sich (8). Andererseits kann man jedes $S \in \text{Gl}(n)$ in der Form $S = S_0 \cdot T$, $T \in \text{Gl}(n)$ schreiben. Aus der Gleichung

$$T^{-1} S_0^{-1} R(A, G) = T^{-1} \begin{bmatrix} \otimes & & & * \\ & \ddots & & \\ & & \otimes & \\ 0 & & & \otimes \\ \hline & & & 0 & \\ & & & & \otimes \end{bmatrix} = \begin{bmatrix} \otimes & & & * \\ & \ddots & & \\ & & \otimes & \\ 0 & & & \otimes \\ \hline & & & 0 & \\ & & & & \otimes \end{bmatrix}$$

folgt aber, daß T die in φ beschriebene Eigenschaft besitzt. Die weiteren Behauptungen ergeben sich aus Kapitel 3.

□

Als Nächstes wird eine äquivalente Formulierung von Definition 1a geliefert. Der Beweis folgt aus Bemerkung 1.

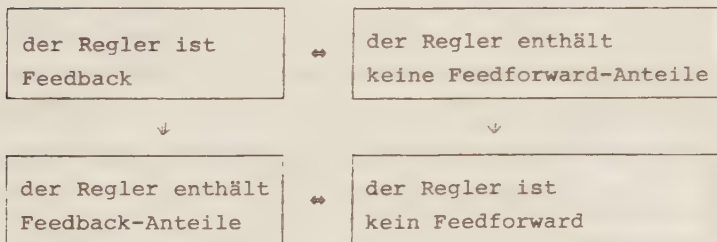
Bemerkung 2

Der Regler (3) ist Feedback bezüglich (3) genau dann, wenn für *alle* $S \in \text{Gl}(n)$ mit (8) die Beziehungen (9) gelten, wobei $(S^{-1}BK, S^{-1}BFCs) \neq 0$. Eine entsprechende Aussage für Feedforward gilt nicht.

Die folgende Aussage klärt den Zusammenhang zwischen Feedback und Feedforward. Der Beweis folgt direkt aus Definition 1 und Bemerkung 2.

Bemerkung 3

Für einen Regler (4) mit $(BK, BFC) \neq 0$ gelten die folgenden Implikationen bezüglich des Systems (3)



Durch Negation erhält man das folgende Diagramm



Weiterhin gelten die folgenden Aussagen.

Bemerkung 4

Obwohl Feedback und Feedforward völlig analog definiert sind, hat Feedback viel angenehmere Eigenschaften als Feedforward. Dieses Phänomen ist in der linearen Systemtheorie jedoch bereits bekannt: Vergleiche die Konzepte der erreichbaren und nicht erreichbaren Teilmenge des Zustandsraumes eines linearen Systems.

Bemerkung 5

Falls $B = G$, so ist jeder Regler mit $BK = 0$ und $BFC \neq 0$ Feedback. In diesem Sinne ist das Kontrollgesetz (1) stets Feedback bezüglich (2). Jeder Regler mit $BK \neq 0$ und $BFC = 0$ ist Feedforward. Dieses deckt sich mit dem üblichen Sprachgebrauch, einen Regler der Form

$$u^k = Kd^k$$

als Feedforward zu bezeichnen.

Bemerkung 6

Erfüllt (4) eine der Bedingungen in Definition 1, so auch

$$u^k = (\alpha F)y^k + (\beta K)d^k,$$

wobei $\alpha, \beta \neq 0$.

Falls die Regler (4), beschrieben durch (F_1, K_1) und (F_2, K_2) , Feedback sind, und

$$(B(K_1+K_2), B(F_1+F_2)C) \neq 0,$$

so ist auch der durch (F_1+F_2, K_1+K_2) beschriebene Regler (4) Feedback. Die Menge aller ein Feedback beschreibenden Tupel (F, K) ist also "fast" ein linearer Raum. Eine entsprechende Aussage für Feedforward gilt nicht.

Bemerkung 7

Eine additive Zerlegung eines beliebigen Reglers in Feedback und Feedforward existiert i.a. nicht. Diese Eigenschaft erhält man mit der Definition von Kalman (Kalman 1972; Dickinson 1976).

Beispiel 1

Es sei $(A, G) \in H(n, 1)$, $C = I$, $B = \beta \cdot e_{k+1} \in \mathbb{R}^n$ mit $\beta \neq 0$ (d.h. $m = 1$), $F \neq 0$ und $K = 0$. Es gilt also

$$A = \begin{bmatrix} h_{11} & \dots & h_{1n} \\ h_{21} & & \vdots \\ \vdots & \ddots & \vdots \\ 0 & & h_{n,n-1} & h_{nn} \end{bmatrix}, \quad G = \begin{bmatrix} h_{10} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \quad \text{und} \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Bigg\}^k \quad (16)$$

mit $h_{i,i-1} \neq 0$ für $i \in \underline{n}$. Jedes Tripel (A,G,B) mit erreichbarem (A,G) und $0 \neq B \in \mathbb{R}^n$ kann per Ähnlichkeit in die obige Form transformiert werden. Schreibt man die Matrix $F \in \mathbb{R}^{1 \times n}$ als

$$F = [F_1, F_2]$$

mit

$$F_1 = [f_1, \dots, f_k] \text{ und } F_2 = [f_{k+1}, \dots, f_n],$$

dann gilt:

- i) der Regler (4) ist Feedback genau dann, wenn $F_1 = 0$;
- ii) der Regler (4) ist Feedforward genau dann, wenn $F_1 \neq 0$ und $f_{k+1} = 0$;
- iii) der Regler (4) besitzt Feedback-Anteile genau dann, wenn $F_1 = 0$ oder $f_{k+1} \neq 0$;
- iv) der Regler (4) besitzt Feedforward-Anteile genau dann, wenn $F_1 \neq 0$.

Wegen Bemerkung 3 folgt iii) und iv) aus i) und ii). Die ersten beiden Behauptungen sieht man aber wie folgt ein:

zu i) Falls $F_1 = 0$, so ist der Regler Feedback nach Definition. Umgekehrt gilt nach Bemerkung 2 für jedes Feedback $F_1 = 0$.

zu ii) Unter den Bedingungen in ii) existiert eine Matrix S der Form

$$S = \begin{bmatrix} I & 0 & * \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{matrix} \} k \\ \} 1 \\ \} n-k-1 \end{matrix}$$

$$\begin{matrix} k & 1 & n-k-1 \end{matrix}$$

so, daß $FS = [F_1, 0]$. Die Gestalt (16) bleibt unter der Ähnlichkeitsaktion mit S erhalten und es gilt

$$S^{-1} BFC S = \begin{bmatrix} 0 & 0 \\ \beta \cdot F_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} \} k \\ \} 1 \\ \} n-k-1 \end{matrix}$$

$\underbrace{\hspace{10em}}_k \quad \underbrace{\hspace{10em}}_{n-k}$

Also ist F Feedforward. Umgekehrt, falls F Feedforward ist, so ist F kein Feedback, also folgt nach i) $F_1 \neq 0$. Ferner impliziert Feedforward, daß BFC nur verschwindende Eigenwerte besitzt, d.h. $f_{k+1} = 0$.

Besitzt das System also beispielsweise das in Bild 7 dargestellte Signalflußdiagramm, so gelten für die Regler, die zu den in den Bildern 8 bis 11 dargestellten Systemen führen, die folgenden Eigenschaften (alle f_i sind ungleich Null):

Bild 8: der Regler ist Feedback

Bild 9, 10: der Regler ist Feedforward

Bild 11: der Regler besitzt Feedback- und Feedforward-Anteile.

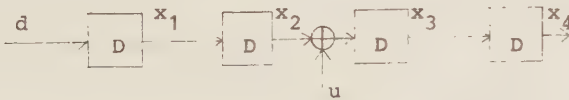


Bild 7

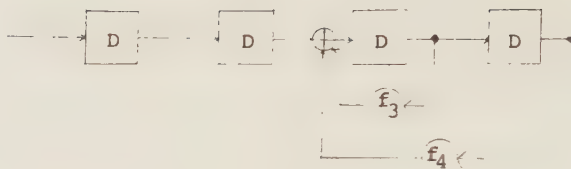


Bild 8

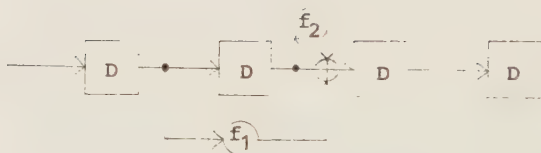


Bild 9

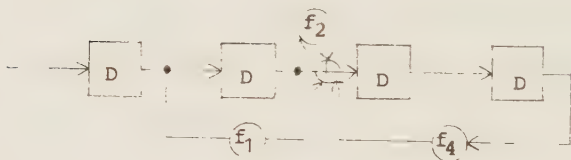


Bild 10

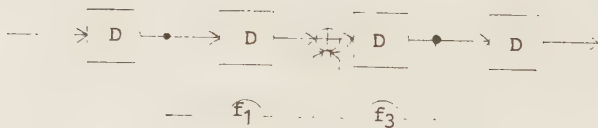


Bild 11

Das in Bild 10 dargestellte System ist aus dem offenen System

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (17)$$

durch Feedforward hervorgegangen, da (17) ähnlich ist zu dem System

$$\tilde{A} = S^{-1}AS = \begin{bmatrix} 0 & 0 & f_4/f_1 & 0 \\ 1 & 0 & 0 & -f_4/f_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad G = S^{-1}G = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad B = S^{-1}B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

mit

$$S = \begin{bmatrix} 1 & 0 & 0 & -f_4/f_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

und $FS = [f_1 \ f_2 \ 0 \ 0]$. In dieser Basis hat das geschlossene System das in Bild 12 dargestellte Signalflußdiagramm. Das Regelgesetz zu Bild 10 ist also Feedforward, da man den "Feedback-Bestandteil" f_4 auch durch reine Ähnlichkeitstransformation erreichen kann.

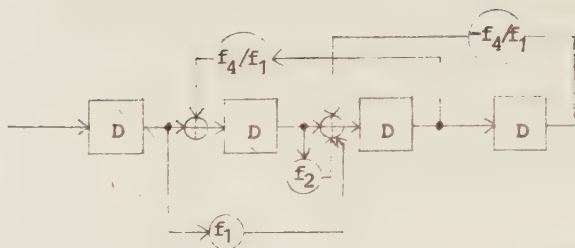


Bild 12

Definition 1 soll jetzt mit Hilfe von *Trajektorien* interpretiert werden. Dazu wird die Lösung von (3) (Trajektorie) zum Anfangswert $x^0 = 0$ und zur Eingangsfunktion $u^k = 0$ ($\forall k \geq 0$) mit

$$x^1(d^0), x^2(d^0, d^1), \dots, x^i(d^0, d^1, \dots, d^{i-1}), \dots$$

bezeichnet. Ein Vektor $x \in \mathbb{R}^n$ heißt in i Schritten erreichbar, falls $d^0, d^1, \dots, d^{i-1}$ existieren mit

$$x = x^i(d^0, d^1, \dots, d^{i-1}).$$

Der Raum aller in i Schritten erreichbaren Zustände $\mathcal{R}^i$ ist bekannterweise gegeben durch

$$\mathcal{R}^i = \text{im}[G, AG, \dots, A^{i-1}G].$$

Zu dem geschlossenen System (5) werden $x_c^i(d^0, d^1, \dots, d^{i-1})$ und $\mathcal{R}_c^i$ genau wie für das offene System definiert.

Feedback läßt sich jetzt wie folgt charakterisieren.

Proposition 1

Für den Regler (4) mit $(BK, BFC) \neq 0$ sind die folgenden Bedingungen äquivalent.

- (i) er ist Feedback bezüglich (3)
- (ii) $\mathcal{R}^i = \mathcal{R}_C^i$ für alle $i \geq 1$
- (iii) $x_C^i(d^0, d^1, \dots, d^{i-1}) - x^i(d^0, d^1, \dots, d^{i-1}) \in \mathcal{R}^{i-1}$
für alle $i \geq 1$ und $d^0, d^1, \dots, d^{i-1}$.
- (iv) $x_C^i(d^0, d^1, \dots, d^{i-1}) - x^i(d^0, d^1, \dots, d^{i-1}) \in \mathcal{R}_C^{i-1}$
für alle $i \geq 1$ und $d^0, d^1, \dots, d^{i-1}$.

Beweis

Es sei (i) erfüllt. Dann gibt es nach Definition 1 ein $S \in \text{Gl}(n)$ mit

$$S^{-1} R(A, G) = \begin{bmatrix} r_1 & & & * \\ & \ddots & & \\ 0 & & r_\ell & \\ \hline & & & 0 \end{bmatrix} \quad (18)$$

und

$$S^{-1} R(A_C, G_C) = \begin{bmatrix} r_1 & & & * \\ & \ddots & & \\ 0 & & r_\ell & \\ \hline & & & 0 \end{bmatrix} \quad (19)$$

Hieraus folgt (ii) unmittelbar. Umgekehrt, falls $S \in \text{Gl}(n)$

irgendeine Matrix mit (18) ist, so ergibt sich (19) aus (ii). Also sind (i) und (ii) äquivalent. Es wird jetzt die Äquivalenz von (ii) und (iii) gezeigt. Die Äquivalenz von (ii) und (iv) ergibt sich dann aus Symmetriegründen. Es sei zunächst (ii) erfüllt und $S \in \text{Gl}(n)$ mit (18) und (19). Wegen

$$x^i(d^0, d^1, \dots, d^{i-1}) = [G, AG, \dots, A^{i-1}G] \begin{bmatrix} d^{i-1} \\ \vdots \\ d^1 \\ d^0 \end{bmatrix}$$

und

$$x_c^i(d^0, d^1, \dots, d^{i-1}) = [G_c, A_c G_c, \dots, A_c^{i-1} G_c] \begin{bmatrix} d^{i-1} \\ \vdots \\ d^1 \\ d^0 \end{bmatrix}$$

folgt hieraus

$$S^{-1} x_c^i(d^0, d^1, \dots, d^{i-1}) - S^{-1} x^i(d^0, d^1, \dots, d^{i-1})$$

$$= \begin{bmatrix} * \\ \vdots \\ * \\ r_i d^0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} * \\ \vdots \\ * \\ r_i d^0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} * \\ \vdots \\ * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Bigg\}^{i-1} \in S^{-1} \mathcal{R}^{i-1},$$

d.h. (iii) ist erfüllt. Andererseits, falls $S \in \text{Gl}(n)$ eine Matrix mit (18) ist und (19) nicht erfüllt ist, so folgt für ein $i \geq 1$

$$s^{-1}x_c^i(1,0,\dots,0) - s^{-1}x^i(1,0,\dots,0) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \}^{i-1}$$

mit $x_2 \neq 0$. Also folgt $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \notin s^{-1} \mathcal{R}^{i-1}$.

□

Proposition 1 läßt sich wie folgt interpretieren.

- Bei Feedback darf es keinen Zustand geben, den man im geschlossenen System "schneller" erreichen kann als im offenen System (kein Vorgriff)
- Bei Feedback darf es keinen Zustand geben, den man im offenen System "schneller" erreichen kann als im geschlossenen System (keine Verschlechterung der Erreichbarkeitseigenschaften)
- Bei Feedback unterscheiden sich die Trajektorien nur um einen Term, der "in der Vergangenheit" liegt, d.h. die "erste Reaktion" auf ein Signal ist im offenen und geschlossenen System gleich.

Für Feedforward ist eine derartige Interpretation nicht möglich.

Es wird jetzt eine weitere Interpretation von Feedback mit Hilfe von speziellen Basen des Zustandsraumes vorgestellt, die sich auch auf Feedforward überträgt. Dazu sind aber einige Vorbereitungen notwendig, die in einem Exkurs zusammengefaßt werden.

Exkurs: Spezielle Basen des Zustandsraumes

Es wird hier das System

$$x^{k+1} = Ax^k + bu^k, \quad x^0 = 0 \quad (20)$$

mit $(A,b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1}$ betrachtet. Für feste $u^0, u^1, \dots, u^{i-1}$ bezeichnet

$$x^1(u^0), x^2(u^0, u^1), \dots, x^i(u^0, u^1, \dots, u^{i-1})$$

die Lösung von (20).

Definition 2

Es sei $(A,b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1}$ und $r_1, \dots, r_n \in \mathbb{R}^n$.

Die Menge $\{r_1, \dots, r_n\}$ heißt *streng durch Kontrolle von (A,b) generierte Basis*, falls $\{r_1, \dots, r_n\}$ eine Basis des $\mathbb{R}^n$ ist und eine Kontrolle $(u^0, u^1, \dots, u^{\ell-1})$ existiert ($\ell := \text{rg } R(A,b)$) mit

$$r_i = x^i(u^0, u^1, \dots, u^{i-1}), \quad i \in \underline{\ell}. \quad (21)$$

Proposition 2

Es sei $R := [r_1, \dots, r_n] \in \mathbb{R}^{n \times n}$. Die Menge $\{r_1, \dots, r_n\}$ ist genau dann eine streng durch Kontrolle von (A,b) generierte Basis, wenn

$$R^{-1}AR = \left[\begin{array}{ccc|ccc} h_{11} & \dots & h_{1\ell} & & & \\ & 1 & 0 & \dots & 0 & h_{2\ell} \\ & & \ddots & & & \vdots \\ & & 1 & \dots & 0 & \vdots \\ & & & \ddots & & \vdots \\ & 0 & & & 1 & h_{\ell\ell} \\ \hline & & & & & 0 \\ & & & & & H_{22} \end{array} \right], \quad R^{-1}b = \left[\begin{array}{c} h_{10} \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{array} \right], \quad h_{10} \neq 0. \quad (22)$$

Beweis

Die Bedingung (21) bedeutet

$$[r_1, \dots, r_\ell] = [b, Ab, \dots, A^{\ell-1}b] \begin{bmatrix} u^0 & u^1 & \dots & u^{\ell-1} \\ & u^0 & \ddots & \vdots \\ & & \ddots & u^1 \\ 0 & & & u^0 \end{bmatrix} \quad (23)$$

Die Vektoren $r_1, \dots, r_\ell$ sind genau dann linear unabhängig, wenn $u^0 \neq 0$. In diesem Fall ist

$$U := \begin{bmatrix} u^0 & u^1 & \dots & u^{\ell-1} \\ & \ddots & \ddots & \vdots \\ & & \ddots & u^1 \\ 0 & & & u^0 \end{bmatrix} \in \text{Gl}(\ell)$$

und die Inverse von U hat die gleiche Struktur wie U :

$$V := U^{-1} = \begin{bmatrix} v^0 & v^1 & \dots & v^{\ell-1} \\ & \ddots & \ddots & \vdots \\ & & \ddots & v^1 \\ 0 & & & v^0 \end{bmatrix}.$$

Multipliziert man (23) von links mit R^{-1} und von rechts mit U^{-1} , so erhält man

$$\begin{matrix} \ell \{ & \overbrace{\begin{bmatrix} V & * \end{bmatrix}}^{\ell \quad n-\ell} \\ \hline n-\ell \{ & \begin{bmatrix} O \end{bmatrix} \end{matrix} = R^{-1} [b, Ab, \dots, A^{n-1}b].$$

Dieses bedeutet aber, daß $R^{-1}AR$ und $R^{-1}b$ die Form (22) haben. □

System-Frobenius-Formen (sowohl solche, bei denen die Parameter in den Zeilen, als auch solche, bei denen die Parameter in den Spalten stehen; vgl. Kapitel 3) entstehen also mit streng durch Kontrolle generierten Basen.

Eine Abschwächung von Definition 2 führt zu

Definition 3

Es sei $(A, b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1}$ und $r_1, \dots, r_n \in \mathbb{R}^n$. Die Menge $\{r_1, \dots, r_n\}$ heißt *durch Kontrolle von (A, b) generierte Basis*, falls $r_1, \dots, r_n$ eine Basis des $\mathbb{R}^n$ ist und Kontrollen

$$u_i = (u_i^0, u_i^1, \dots, u_i^{i-1}), \quad i \in \underline{\ell},$$

existieren mit

$$r_i = x^i (u_i^0, u_i^1, \dots, u_i^{i-1}), \quad i \in \underline{\ell}. \quad (24)$$

Hierbei ist $\ell := \text{rg } R(A, b)$.

Proposition 3

Es sei $R := [r_1, \dots, r_n]$. Die Menge $\{r_1, \dots, r_n\}$ ist genau dann eine durch Kontrolle von (A, b) generierte Basis, wenn

$$R^{-1}AR = \left[\begin{array}{ccc|cc} h_{11} & \dots & h_{1\ell} & & \\ h_{21} & \ddots & \vdots & & \\ & \ddots & \ddots & & \\ 0 & & h_{\ell, \ell-1} & h_{\ell\ell} & \\ \hline & & 0 & & \end{array} \right] \begin{array}{c} H_{12} \\ \\ \\ \\ H_{22} \end{array}, \quad R^{-1}b = \left[\begin{array}{c} h_{10} \\ 0 \\ \vdots \\ 0 \\ \hline 0 \end{array} \right], \quad (25)$$

wobei $h_{i, i-1} \neq 0$ für $i \in \underline{\ell}$.

Beweis

Die Bedingung (24) bedeutet

$$[r_1, \dots, r_\ell] = [b, Ab, \dots, A^{\ell-1}b] \begin{bmatrix} u_1^0 & u_2^1 & \dots & u_\ell^{\ell-1} \\ & u_2^0 & & \vdots \\ & & \ddots & u_\ell^0 \\ 0 & & & \ddots & u_\ell^0 \end{bmatrix}, \quad (26)$$

und daher ist $\{r_1, \dots, r_n\}$ eine durch Kontrolle von (A, b) generierte Basis genau dann, wenn

$$R^{-1}[b, Ab, \dots, A^{n-1}b] = \begin{bmatrix} \oplus & & & * \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \oplus & * \\ \hline & & & & 0 \end{bmatrix} \quad (27)$$

Dieses ist äquivalent zu (25).

□

System-Hessenberg-Formen (vgl. Kapitel 3) entstehen also mit durch Kontrolle generierten Basen.

Für multivariable Systeme gibt es verschiedene Möglichkeiten, durch Kontrolle generierte Basen zu definieren. Fordert man beispielsweise, daß für alle $j \in \underline{m}$ die ersten ℓ_j Vektoren ($\ell_j := \text{rg } R(A, [b_1, \dots, b_j])$) nur durch die ersten j Eingänge erzeugt werden, so erhält man Zustandsdarstellungsformen, die der Hermite- bzw. Hermite-Hessenberg-Form (vgl. Kapitel 3) ähnlich sehen. Man kann jedoch auch durch Kontrolle generierte Kronecker-Basen definieren. Die Details werden hier nicht ausgearbeitet. Statt dessen wird kurz auf das duale Konzept zu durch Kontrolle generierten Basen eingegangen.

Dazu sei das System gegeben durch

$$x^{k+1} = Ax^k \quad (28)$$

wobei $(A, c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{1 \times n}$.

Ein Vektor $x^0 \in \mathbb{R}^n$ heißt *unsichtbar vom Grade i* ($i = 0, 1, 2, \dots, n-1$), falls für die zum Anfangszustand x^0 gehörende Lösung von (28) gilt:

$$y^0 = y^1 = \dots = y^{i-1} = 0. \quad (29)$$

Für $i = 0$ ist die Bedingung (29) per Definition leer. x^0 ist also unsichtbar vom Grade i genau dann, wenn

$$x^0 \in \bigcap_{j=0}^{i-1} \ker (cA^j).$$

Definition 4

Es sei $(A, c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{1 \times n}$ und $r_1, \dots, r_n \in \mathbb{R}^n$.

Die Menge $\{r_1, \dots, r_n\}$ heißt *maximal bezüglich (A, c) unsichtbare Basis*, falls $\{r_1, \dots, r_n\}$ eine Basis ist und r_i unsichtbar vom Grade $i-1$ ist, $i \in \underline{n}$.

Proposition 4

Es sei $R = [r_1, \dots, r_n]$. Die Menge $\{r_1, \dots, r_n\}$ ist genau dann eine maximal bezüglich (A, c) unsichtbare Basis, wenn

$$R^{-1}AR = \left[\begin{array}{ccc|ccc} h_{11} & h_{12} & & & & \\ \vdots & \ddots & & & & \\ & & h_{\ell-1, \ell} & & & \\ h_{\ell, 1} & \dots & h_{\ell \ell} & & & \\ \hline & & & H_{21} & & \\ & & & & H_{22} & \end{array} \right], \quad (30)$$

$$cR = [h_{01} \ 0 \ \dots \ 0 \ | \ 0].$$

wobei $h_{i-1,i} \neq 0$ für $i \in \underline{\ell}$ und

$$\ell := n - \dim \left(\bigcap_{j=0}^{n-1} \ker (cA^j) \right) = \operatorname{rg} \begin{bmatrix} c \\ cA \\ \vdots \\ cA^{n-1} \end{bmatrix}. \quad (31)$$

Beweis

Eine Basis $\{r_1, \dots, r_n\}$ ist genau dann eine maximal bezüglich (A, c) unsichtbare Basis, wenn

$$\begin{bmatrix} c \\ cA \\ cA^2 \\ \vdots \\ cA^{n-1} \end{bmatrix} [r_1, \dots, r_n] = \begin{bmatrix} y_{11} & & 0 & & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ y_{\ell 1} & \dots & y_{\ell \ell} & & \\ \hline & & & y_{\ell+1, \ell+1} & 0 \\ * & & & \vdots & \\ & & & y_{n, \ell+1} & \dots & y_{nn} \end{bmatrix}.$$

Aufgrund der Definition von ℓ gilt $y_{ii} \neq 0$ für $i \in \underline{\ell}$ und

$$\begin{bmatrix} y_{\ell+1, \ell+1} & & 0 \\ \vdots & \ddots & \vdots \\ y_{n, \ell+1} & \dots & y_{nn} \end{bmatrix} = 0.$$

□

Hieraus folgt die Behauptung

Definition 2 und Proposition 2 lassen sich genauso dualisieren.

Der Exkurs wird abgeschlossen durch die folgende äquivalente Formulierung von Satz 1 in Kapitel 3.

Proposition 5

Es sei $(A, b, c) \in \Sigma(n, 1, 1)$. Dann gibt es eine Basis $\{r_1, \dots, r_n\}$, die gleichzeitig durch Kontrolle von (A, b) generiert ist und maximal bezüglich (A, c) unsichtbar ist genau dann, wenn $(A, b, c) \in \hat{\Sigma}(n, 1, 1)$. Bis auf Normierung ist diese Basis eindeutig bestimmt.

(Ende des Exkurses)

Im Hinblick auf den Exkurs gilt die folgende Charakterisierung von Feedback und Feedforward.

Proposition 6

Es sei (A, G) erreichbar und ein Regler (4) mit $BK = 0$, $BFC \neq 0$ gegeben.

- i) Der Regler (4) ist genau dann Feedback bezüglich (3), wenn eine durch Kontrolle von (A, G) generierte Basis $\{r_1, \dots, r_n\}$ existiert mit

$$A_c r_i - A r_i \in \text{span}\{r_1, \dots, r_i\}, \quad i \in \underline{n}.$$

- ii) Der Regler (4) ist genau dann Feedforward bezüglich (3), wenn eine durch Kontrolle von (A, G) generierte Basis $\{r_1, \dots, r_n\}$ existiert mit

$$A_c r_i - A r_i \in \text{span}\{r_{i+1}, \dots, r_n\}.$$

Dabei ist für $i = n$ per Definition $\text{span}\{r_{i+1}, \dots, r_n\} = \{0\}$.

Beweis

Nach Proposition 3 ist $\{r_1, \dots, r_n\}$ genau dann eine durch Kontrolle von (A, G) generierte Basis, wenn mit $S := [r_1, \dots, r_n] \in \text{Gl}(n)$ die Gleichung (8) in Definition 1 erfüllt ist. Betrachtet man jetzt die Gleichungen

$$\begin{aligned} G &= S(S^{-1}G), \quad AS = S(S^{-1}AS) \\ G_C &= S(S^{-1}G_C), \quad A_C S = S(S^{-1}A_C S) \end{aligned}$$

spaltenweise, so ergibt sich die Behauptung. □

Es werden jetzt zwei Anwendungsfälle betrachtet, in denen typischerweise Feedforward eingesetzt wird.

Zunächst wird Heymann's Lemma (Heymann 1968; Hautus 1977) untersucht, das auch wie folgt formuliert werden kann.

Heymann's Lemma

Es sei das System (3) gegeben mit $C = I$, $q = 1$ und $(A, [B, G])$ erreichbar. Dann existiert ein Regler der Form (4) mit $K = 0$, so daß das System

$$(A + BF, G) \tag{32}$$

erreichbar ist.

Es sei u^k zunächst skalar, d.h. das System habe zwei skalare Eingänge. Dann gilt:

Proposition 7

Es sei zusätzlich zu den Voraussetzungen von Heymann's Lemma

noch $m = 1$, und (A, G) sei nicht erreichbar, d.h.

Heymann's Lemma sei nicht trivial erfüllt. Dann gilt:

- Falls $(A+BF, G)$ erreichbar ist, so enthält der Regler (4) (mit $K = 0$) Feedforward-Anteile.
- Es gibt ein Feedforward der Form (4) (mit $K = 0$), so daß $(A+BF, G)$ erreichbar ist.

Beweis

Falls $(A+BF, G)$ erreichbar ist, so folgt

$$\varphi_G^{n-1} \neq 0,$$

also kann der Regler nach Proposition 1 kein Feedback sein und nach Bemerkung 3 enthält er Feedforward-Anteile.

Um den zweiten Teil zu zeigen, wähle $S \in \text{Gl}(n)$, so daß das System Hermite-Form (Hinrichsen und Prätzel-Wolters 1983) annimmt:

$$S^{-1}AS = \left[\begin{array}{ccc|ccc} 0 & & & x & & x \\ 1 & & 0 & \vdots & & \vdots \\ & \ddots & & \vdots & 0 & \vdots \\ 0 & & 1 & x & & x \\ \hline & & & 0 & 0 & x \\ & & & 1 & \ddots & \vdots \\ & 0 & & 0 & 1 & x \end{array} \right], \quad S^{-1}G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (33)$$

Aus (33) kann man ersehen, daß

$$(S^{-1}AS + S^{-1}B \tilde{F}, S^{-1}G)$$

erreichbar ist, wobei

$$\tilde{F} = [0 \dots 0 \ 1 \quad 0 \quad] .$$

Der Regler (4) mit $K = 0$ und $F = \tilde{F} S^{-1}$ ist Feedforward.

□

Eine direkte Verallgemeinerung des vorherigen Beweises ergibt das folgende Resultat für mehr als zwei Eingänge.

Proposition 8

Zusätzlich zu den Voraussetzungen von Heymann's Lemma sei (A, G) nicht erreichbar. Schreibe $B = [b_1, \dots, b_m]$.

Dann gilt:

- Falls $(A + BF, G)$ erreichbar ist, so enthält der Regler (4) (mit $K = 0$) Feedforward-Anteile.
- Es gibt einen Regler der Form (4) mit $K = 0$ und

$$F = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} ,$$

so daß $(A + BF, G)$ erreichbar ist, und für alle $i \in \underline{m}$ durch f_i ein Feedforward bezüglich des Systems

$$x^{k+1} = (A + [b_1, \dots, b_{i-1}] \begin{bmatrix} f_1 \\ \vdots \\ f_{i-1} \end{bmatrix}) x^k + b_i u_i^k + G d^k$$

beschrieben ist.

Die Regler zu $u_1, \dots, u_m$ werden also sukzessiv bestimmt, und jeder einzelne Regler ist Feedforward bezüglich des bereits konstruierten geschlossenen Systems.

Als zweiter Anwendungsfall wird das Störungsentkopplungsproblem durch Zustandsrückführung (vgl. Abschnitt 5.1.1) untersucht. Dazu wird Definition 1 aber zunächst dualisiert.

Definition 5

Es sei das System

$$\begin{aligned}x^{k+1} &= Ax^k + Bu^k \\y^k &= Cx^k \\z^k &= Hx^k\end{aligned}\tag{34}$$

mit $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ und $H \in \mathbb{R}^{1 \times n}$ gegeben. Der Regler

$$u^k = Fy^k\tag{35}$$

heißt *Feedback (Feedforward)* bzw. enthält *Feedback-Anteile (Feedforward-Anteile)* bezüglich (34), wenn der Regler

$$u^k = F^T y^k\tag{36}$$

Feedback (Feedforward) ist bzw. Feedback-Anteile (Feedforward-Anteile) besitzt bezüglich des Systems

$$\begin{aligned}x^{k+1} &= A^T x^k + C^T u^k + H^T d^k \\y^{k+1} &= B^T x^k.\end{aligned}\tag{37}$$

Proposition 9

Es sei das System

$$\begin{aligned}x^{k+1} &= Ax^k + Bu^k + Gd^k \\y^k &= Cx^k, \quad C = I \\z^k &= Hx^k\end{aligned}$$

gegeben mit gleichen Dimensionen wie in (34) und $G \in \mathbb{R}^{n \times q}$.
 Ferner sei $HA^i G \neq 0$ für ein $i \geq 0$, d.h. das Störungsentkopplungsproblem sei nichttrivial. Dann gilt:

- Falls der Regler (35) das Störungsentkopplungsproblem SEPZ (vgl. Abschnitt 5.1.1) löst, so enthält er Feedforward-Anteile bezüglich (34).
- Falls das Störungsentkopplungsproblem SEPZ lösbar ist, so ist es durch Feedforward bezüglich (34) lösbar.

Beweis

Falls der Regler (35) das Störungsentkopplungsproblem löst, so gilt $HA_C^i G = 0$ für alle $i \geq 0$. Da das Störungsentkopplungsproblem nichttrivial ist, folgt $\mathcal{R}^{n-1} \neq \mathcal{R}_C^{n-1}$, und Proposition 1 und Bemerkung 3 implizieren die erste Behauptung.

Falls nun SEPZ lösbar ist, so gibt es (vgl. Kapitel 5) ein $S \in \text{Gl}(n)$ mit

$$S^{-1}AS = \begin{bmatrix} * & & * \\ \hline & * & * \\ 0 & \text{Diagonal} & \\ & 0 & * \end{bmatrix}, \quad S^{-1}B = \begin{bmatrix} * \\ * \\ \hline b \\ 0 \end{bmatrix} \left\{ r_1 \right., \quad S^{-1}G = \begin{bmatrix} * \\ * \\ \hline 0 \\ 0 \end{bmatrix},$$

$$HS = \begin{bmatrix} 0 & 0 \dots 0 & \text{Diagonal} \end{bmatrix},$$

wobei b ein nichtverschwindender Zeilenvektor ist. Ohne Einschränkung sei der erste Eintrag von b ungleich Null. Mittels einer Ähnlichkeitstransformation mit $T \in \text{Ut}^*(n)$

kann man immer erreichen, daß alle Einträge in der ersten Spalte von B "über b " verschwinden. Ohne Einschränkung kann man also auch annehmen, daß

$$S^{-1}B = \left[\begin{array}{c|c} 0 & * \\ \hline 0 & * \\ \hline 0 & * \\ \hline 0 & 0 \end{array} \right] \begin{array}{l} r_1 \\ \\ \\ \end{array}.$$

Jetzt sieht man aber, daß ein Feedforward bezüglich (34), gegeben durch

$$F = \left[\begin{array}{ccccccc} \overbrace{0 \dots 0}^{r_1} \alpha & 0 & \dots & 0 \\ & 0 & & \end{array} \right] S^{-1},$$

das Störungsentkopplungsproblem löst.

□

Als multivariablen Analogon von Definition 1 (d^k nicht notwendig skalar) wird hier vorgeschlagen.

Definition 6

Es sei ein System (3) und ein Regler (4) gegeben.

(a) Der Regler (4) heißt *Feedback* bezüglich (3), falls

$S \in Gl(n)$ existiert mit

$$S^{-1}G = \begin{bmatrix} H_{10} \\ \\ \\ 0 \end{bmatrix}, \quad S^{-1}AS = \begin{bmatrix} H_{11} & \cdot & \cdot & \cdot & \cdot & \cdot & H_{1,l+1} \\ & H_{21} & & & & & \vdots \\ & & \ddots & & & & \vdots \\ & & & H_{l,l-1} & H_{ll} & & \vdots \\ & & & & 0 & H_{l+1,l+1} & \end{bmatrix}, \quad (38)$$

wobei $H_{i,j} \in \mathbb{R}^{l_i \times l_j}$ und $H_{i,i-1}$ vollen Zeilenrang besitzt für alle $i \in \underline{l+1}$, $j = 0, 1, \dots, l+1$, und

$$S^{-1}BK = \begin{bmatrix} \\ \\ \\ 0 \end{bmatrix}, \quad S^{-1}BFCS = \begin{bmatrix} M_{11} & \cdot & \cdot & \cdot & \cdot & \cdot & M_{1,l+1} \\ & \ddots & & & & & \vdots \\ & & \ddots & & & & \vdots \\ & & & 0 & & & \vdots \\ & & & & \ddots & & \vdots \\ & & & & & M_{l+1,l+1} & \end{bmatrix}, \quad (39)$$

wobei $M_{i,j} \in \mathbb{R}^{l_i \times l_j}$ für $i, j \in \underline{l+1}$ und $(S^{-1}BK, S^{-1}BFCS) \neq 0$.

- (b) Der Regler (4) heißt *Feedforward* bezüglich (3), falls $S \in \text{Gl}(n)$ existiert mit (38) und

$$S^{-1}BK = \begin{bmatrix} M_{10} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ M_{l+1,0} \end{bmatrix}, \quad S^{-1}BFCS = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ M_{21} & & & & & & \vdots \\ \cdot & \ddots & & & & & \vdots \\ \cdot & & \ddots & & & & \vdots \\ \cdot & & & \ddots & & & \vdots \\ M_{l+1,1} & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix},$$

wobei $M_{ij} \in \mathbb{R}^{l_i \times l_j}$ für $i, j = 0, 1, \dots, l+1$ und $(S^{-1}BK, S^{-1}BFCS) \neq 0$.

Feedback-Anteile und Feedforward-Anteile kann man analog definieren. Die Details des multivariablen Falls werden hier nicht ausgeführt. Es soll lediglich angemerkt werden, daß sich für multivariables z Proposition 10 derart verallgemeinern läßt, daß jeder störungsentkoppelnde Regler Feedforward-Anteile besitzt und das Störungsentkopplung immer möglich ist durch eine Summe von Feedforwards. Dieses kann man beispielsweise mit Hilfe von Bild 3 aus Kap. 5 überprüfen.

Eine mögliche Verallgemeinerung von Definition 1 (bzw. Definition 7) auf dynamische Regler der Form

$$\begin{aligned} z^{k+1} &= Mz^k + Ny^k \\ u^k &= Ez^k + Fy^k \end{aligned} \quad (40)$$

besteht darin, die Bedingungen an BFC zu ersetzen durch Bedingungen an BP_iC , wobei $P_{-1}, P_0, P_1, \dots$ die Markovparameter von (40) sind:

$$P_{-1} = F, \quad P_i = EM^iN, \quad i \geq 0.$$

7. LITERATUR

- ACKERMANN, J., 1977,
Entwurf durch Polvorgabe, Regelungstechnik, Heft 6,
S. 173-179 und Heft 7, S.209-215.
- APLEVICH, J.D., 1974,
Direct computation of canonical forms for linear
systems by elementary matrix operations,
IEEE Trans.Automat.Contr., Vol.AC-19,S.124-126.
- APLEVICH, J.D., 1979,
Tableau methods for analysis and design of linear
systems , Automatica, Vol. 15, S.419-429.
- BHATTACHARYYA, S.P., und DE SOUZA, E., 1981,
Pole assignment via Sylvesters' equation,
Systems & Control Letters, Vol.1, No.3.
- BHATTACHARYYA, S.P., DEL NERO GOMES, A.C., und J.W.HOWZE, 1983,
The structure of robust disturbance rejection control,
IEEE Trans.Automat.Contr., Vol.AC-28, No.9, S.874-881.
- BOSGRA, O.K., und van der WEIDEN, A.J.J., 1980,
Input-output invariants for linear multivariable systems,
IEEE Trans.Automat.Contr., Vol.AC-25, No.1, S.20-36.
- BYRNES, C.I., und DUNCAN, T.E., 1982,
On certain topological invariants arising in system theory,
in: New Directions in Applied Mathematics, P.J.Hilton,
G.S.Young (ed.) (Springer, New York-Heidelberg-Berlin),
S.29-71.
- BYRNES, C.I., und LINDQUIST, A., 1982,
The stability and instability of partial realizations,
Systems & Control Letters, Vol.2, S. 99-105.
- CAVIN, R.K., und BHATTACHARYYA, S.P., 1982,
Robust and well conditioned eigenstructure assignment
via Sylvester's equation, in: Proc.1982 Am.Contr.Conf.,
Arlington.
- DICKINSON, B.W., 1976,
The classification of linear state variable control
laws, Siam J.Control and Optimization, Vol.14, No.3,
S.467-477.
- DONGARRA, J.J., MOLER, C.B., BUNCH, J.R., STEWART, G.W., 1979,
LINPACK User's Guide, SIAM, Philadelphia.
- EISING, R., 1982,
Pole assignment, a new proof and algorithm, Systems &
Control Letters, Vol.2, No.1, S.6-12.

- EISING, R., 1983,
A collection of numerically reliable algorithms for
the deadbeat control problem, Memorandum COSOR 83-13,
Eindhoven University of Technology, Dept. of Mathematics
and Computing Science.
- EISING, R., 1984,
Between controllable and uncontrollable,
Dept. of Mathematics and Computing Science, Eindhoven
University of Technology.
- GANTMACHER, F. R., 1970,
Matrizenrechnung I und II (VEB Deutscher Verlag der
Wissenschaften, Berlin).
- GRAGG, W. B., und LINDQUIST, A., 1983,
On the partial realization problem, Linear Algebra
and its Applications, Vol. 50, S. 277-319.
- HAUTUS, M. L. J., 1977,
A simple proof of Heymann's Lemma, IEEE Trans. Automat.
Contr., Vol. AC-22, No. 5, S. 885-886.
- HEYMANN, M., 1968,
Comments on "Pole assignment in multi-input control-
lable linear systems", IEEE Trans. Automat. Contr.,
Vol. AC-13, S. 748-749.
- HINRICHSSEN, D., 1984,
A canonical form for multinomial systems, erscheint in:
Proc. III IFIP Working Conf. on System Modelling and
Optimization, Hanoi 1983, Lecture Notes in Control
and Information Sciences.
- HINRICHSSEN, D., und LINNEMANN, A., 1984,
Normalformen vom Hermite-Typ und die Berechnung
dominanter Hermite-Indizes strukturierter Systeme,
Regelungstechnik, 32. Jahrgang, Heft 4, S. 124-130.
- HINRICHSSEN, D., MÜNZNER, H. F., und PRÄTZEL-WOLTERS, D., 1981,
Parametrization of (C,A)-invariant subspaces, Systems
& Control Letters, Vol. 1, No. 3, S. 192-199.
- HINRICHSSEN, D., und PRÄTZEL-WOLTERS, D., 1983,
Generalized Hermite matrices and complete invariants
of (strict) system equivalence, SIAM J. Control and
Optimization, Vol. 21, S. 289-305.
- HINRICHSSEN, D., und PRÄTZEL-WOLTERS, D., 1984,
A canonical form for static linear output feedback, in:
Mathematical Theory of Networks and Systems,
P. A. Fuhrmann (ed.), Lecture Notes in Control and
Information Sciences 58, S. 441-462.
- KALMAN, R. E., 1963,
Liapunov functions for the problem of Lure in automatic
control, Proc. Nat'l Acad. of Sci. (USA), Vol. 40, S. 201-205.

- KALMAN, R.E., 1971,
On minimal partial realizations of a linear input/
output map, in: Aspects of Network and System Theory,
R.E.Kalman and N.De Claris (eds), Holt, Rinehart and
Winston, S.385-408.
- KALMAN, R.E., 1972,
Kronecker invariants and feedback, Ordinary Differential
Equations, L.Weiss (ed.), Academic Press, New York).
- KAMEN, E.W., 1978,
Lectures on algebraic system theory: linear systems
over rings, NASA Contractor Report 3016.
- KATTI, S.K., 1983,
Pole placement in multi-input systems via elementary
transformations, Int.J.Control, Vol.37, No.2, S.315-347.
- KAUTSKY, J., und NICHOLS, N.K., 1983,
Robust eigenstructure assignment,
Numerical Analysis Rpt. NA/2/83, University of Reading.
- KAUTSKY, J., NICHOLS, N.K., VAN DOOREN, P., und FLETCHER, L., 1982,
Numerical methods for robust eigenstructure assignment
in control system design, in: Proc.Workshop Numerical
Treatment of Inverse Problems for Differential and
Integral Equations, Heidelberg, (Birkhäuser, Boston)
- KLEMA, V.C., und LAUB, A.J., 1980,
The singular value decomposition: its computation and some
applications, IEEE Trans. Automat. Contr., Vol. AC-25,
No.2, S.164-176.
- KONSTANTINOV, M.M., PETKOV, P.Hr., und CHRISTOV, N.D., 1980,
Synthesis of linear systems with desired equivalent
form, J.Comp. & Appl.Math., Vol.6, S.27-35.
- KONSTANTINOV, M.M., PETKOV, P.H., und CHRISTOV, N.D., 1981,
Invariants and canonical forms for linear multivariable
systems under the action of orthogonal transformation
groups, Kybernetika, Vol.17, No.5, S.413-424.
- KONSTANTINOV, M.M., PETKOV, P.H., und CHRISTOV, N.D., 1981a,
A Schur approach to pole assignment problem, Control and
Cybernetics, Vol.10, No.1-2, S.19-29.
- LAUB, A.J., 1983,
Numerical aspects of control design computations, to appear.
- LAUB, A.J., und LINNEMANN, A., 1984,
Hessenberg forms in linear systems theory, in preparation.
- LIN, C.T., 1974,
Structural Controllability, IEEE Trans.Automat.Contr.,
Vol.AC-19, S.201-208.

- LINNEMANN, A., 1984,
Decentralized stabilization by high gain feedback,
Forschungsschwerpunkt Dynamische Systeme, Report
Nr.115; erscheint in: Proc.First European Workshop
on the Real Time Control of Large Scale Systems,
Patras, July 9-12, 1984 (Springer, Berlin-Heidelberg-
New York).
- LUENBERGER, D.G., 1967,
Canonical forms for linear multivariable systems,
IEEE Trans.Automat.Contr., Vol.AC-12, S.290-293.
- MIMINIS, G.S., und PAIGE, C.C., 1982,
An Algorithm for pole assignment of time invariant
linear Systems, Int.J.Control, Vol.35, No.2, S.341-354.
- MIMINIS, G.S., und PAIGE, C.C., 1982a,
An algorithm for pole assignment of time invariant
multi-input linear systems, in: Proc. 21st CDC, S.62-67.
- MOORE, B.C., 1976,
On the flexibility offered by state feedback in
multivariable systems beyond closed loop eigenvalue
assignment, IEEE Trans.Automat.Contr., S.689-692.
- MUNRO, N., und MCLEOD, R.S., 1971,
Minimal realization of transfer-function matrices
using the system matrix, Proc.IEE, Vol.118, No.9,
S.1298-1301.
- NOUR ELDIN, H.A. 1974,
Polynomfestlegung für Mehrgrößenregelsysteme,
Regelungstechnik, Heft 9, S.286-287.
- NOUR ELDIN, H.A., 1974a,
Polynomfestlegung beim Beobachterentwurf, Regelungs-
technik, Heft 9, S.288-289.
- NOUR ELDIN, H.A., 1977,
Minimalrealisierung der Matrix-Übertragungsfunktion,
Regelungstechnik, Heft 3, S.82-87.
- NOUR ELDIN, H.A., und HEISTER, M., 1980,
Zwei neue Zustandsdarstellungsformen zur Gewinnung von
Kroneckerindizes, Entkopplungsindizes und eines Prim-
Matrix-Produktes, Regelungstechnik, 28.Jahrgang, Heft
12, S.420-425 und 29.Jahrgang, Heft 1, S.26-30.
- PAIGE, C.C., 1981,
Properties of numerical algorithms related to computing
controllability, IEEE Trans.Automat.Contr., Vol.AC-26,
No.1, S.130-138.
- PATEL, R.V., 1981,
Computation of matrix fraction descriptions of linear
time-invariant systems, IEEE Trans.Automat.Contr., Vol.
AC-26, No.1, S.148-161.

- PATEL, R.V., 1981a,
Computation of minimal-order state-space realizations
and observability indices using orthogonal trans-
formations, Int.J.Control, Vol.33, No.2, S.227-246.
- PETKOV, P.H., KONSTANTINOV, M.M., und CHRISTOV, N.D., 1982,
Pole assignment by orthogonal triangularization, Proc.
11-th Spring Conf. of the Union of Bulgarian Mathematicians,
Burgas, S.265-273.
- POPOV, V.M., 1972,
Invariant description of linear time - invariant controll-
able systems, SIAM J.Control, Vol.10, S.252-264.
- ROSENBROCK, H.H., 1970,
State Space and Multivariable Theory (Nelson, London).
- RUHE, A., 1977,
Computation of eigenvalues and eigenvectors, in:
Sparse Matrix Techniques, V.A.Barker (ed.), Lecture
Notes in Mathematics 572, S.130-184.
- SEZER, M.E., und HÜSEYİN, Ö., 1978,
Stabilization of linear time-invariant interconnected
systems using local state feedback, IEEE Trans. Systems
Man Cybernetics, Vol.SMC-8, No.10, S.751-756.
- SIMA, V., 1981,
An efficient Schur method to solve the stabilizing
problem, IEEE Trans. Automat. Contr., AC-26, 724-725.
- SMITH, B.T., BOYLE, J.M., DONGARRA, J.J., GARBOW, B.S., IKEBE, Y.,
KLEMA, V.C., und MOLER, C.B., 1976,
Matrix Eigensystem Routines - EISPACK Guide, Lecture
Notes in Computer Science 6, (Springer, Berlin-Heidelberg-
New York).
- SONTAG, E.D., 1981,
Linear systems over commutative rings: a (partial)
updated survey, Proc. IFAC World Congress 1981, Kyoto.
- STEWART, G.W., 1973,
Introduction to matrix computations,
(Academic Press, New York - San Francisco - London).
- STOER, J., und BULIRSCH, R., 1976/1978,
Einführung in die numerische Mathematik I und II,
(Springer, Berlin-Heidelberg-New York).
- TSE, E.C.Y., MEDANIC, J., und PERKINS, W.R., 1978,
Generalized Hessenberg transformations for reduced order
modelling of large scale systems, Int. J. Control, Vol.
27, pp.493-512.

- VAN DOOREN, P.M., 1981,
The generalized eigenstructure problem in linear
system theory, IEEE Trans. Automat. Contr., Vol. AC-26, No. 1,
S. 111-129.
- VAN DOOREN, P.M., 1984,
Deadbeat control: a special inverse eigenvalue problem,
in: Mathematical Theory of Networks and Systems, P.A.
Fuhrmann (ed.), Lecture Notes in Control and Information
Sciences 58.
- VARGA, A., 1979,
Numerically reliable algorithm to test controllability
Electronics Letters, Vol. 15, No. 15, S. 452-453.
- VARGA, A., 1981,
Numerically stable algorithm for standard controllability
form determination, Electronics Letters, Vol. 17, No. 2, S. 74-75.
- VARGA, A., 1981a,
A Schur method for pole assignment, IEEE Trans. Automat.
Contr., Vol. AC-26, No. 2, S. 517-519.
- VARGA, A. und SIMA, V., 1981,
Numerically stable algorithm for transfer function matrix
evaluation, Int. J. Control, Vol. 33, No. 6, S. 1123-1133.
- WALL, H.S., 1967,
Analytic Theory of Continued Fractions, (Chelsea Publishing,
New York).
- WERNER, W., 1983,
A generalized companion matrix of a polynomial and some
applications, Linear Algebra and its Applications, Vol.
55, S. 19-36.
- WILKINSON, J.H., 1961,
Error analysis of direct methods of matrix inversion,
J. Ass. comp. Mach., Vol. 8, S. 281-330.
- WILKINSON, J.H., 1965,
The algebraic eigenvalue problem, (Clarendon Press, Oxford).
- WILKINSON, J.H., und REINSCH, C., 1971,
Handbook for Automatic Computation II - Linear Algebra,
(Springer, Berlin - Heidelberg - New York).
- WILLEMS, J.C., 1981,
Almost invariant subspaces: An approach to high gain
feedback design - Part I: Almost controlled invariant
subspaces, IEEE Trans. Automat. Contr., Vol. AC-26, No. 1,
S. 235-252.
- WILLEMS, J.C., 1982,
Almost invariant subspaces: An approach to high gain
feedback design - Part II: Almost conditionally invariant
subspaces, IEEE Trans. Automat. Contr., Vol. AC-27,
No. 5, S. 1071-1085.

- WILLEMS, J.C., und COMMAULT, C., 1981,
Disturbance decoupling by measurement feedback with
stability or pole placement, SIAM J. Control and Optimization,
Vol. 19, No. 4, S. 490-504.
- WILLEMS, J.C. und Ikeda, M., 1984,
Decentralized stabilization of large-scale interconnected
systems, erscheint in: Proc. Sixth International Con-
ference on Analysis and Optimization of Systems, Nice,
June 19-22, 1984.
- WOLOVICH, W.A., 1974,
Linear Multivariable Systems, (Springer, New York-
Heidelberg-Berlin).
- WONHAM, W.M., 1974,
Linear Multivariable Control - A Geometric Approach,
Lecture Notes in Economics and Mathematical Systems
Nr. 101, (Springer, Berlin-Heidelberg-New York).

ANHANG: Schlüssel zum Auffinden von Textstellen

Einige Algorithmen wurden im Text nicht systematisch numeriert, sondern mit Buchstabenfolgen bezeichnet. Die folgende Aufstellung gibt die Seiten an, auf denen diese Algorithmen im Text zu finden sind.

Algorithmus	SHF	S.	32
"	SHFM	S.	66
"	SJF	S.	89
"	PZF	S.	95
"	PZN	S.	97
"	PZS	S.	98
"	PZEB	S.	104
"	PZSM	S.	114
"	PZDB	S.	125
"	MAIU	S.	145
"	MAIU	S.	180

Im Kapitel 5 werden vier Störungsentkopplungsprobleme behandelt. Die folgende Tabelle gibt eine Übersicht über die Seiten, auf denen die einzelnen Probleme behandelt werden.

	SEPZ	SEPA	SEPPZ	FSEPSSZ
Definition des Problems u. geometrische Theorie	133	135	137	139
Algorithmen zur Überprüfung der Lösbarkeit (Analyse)	157	158	165	166
Algorithmen zur Lösung (Synthese)	168	169	172	178
Charakterisierungen der Lösbarkeit	204	209	215	219

